

ADDING TO THE REALISM OF YOUR AIRPORT

A Tutorial for Microsoft Flight Simulators FS9 and FSX (“MFS”)

Using an airport editor such as ADE in association with the SDK tools, it is a relatively simple task to modify a stock airport so that it more closely resembles its real-world counterpart by adding parking and aprons – or even to create a new airport from scratch. But, the result often won’t look much like the real thing when taxiing about in the user aircraft. In particular:

- the rounded “fillets” drawn at each corner where a taxiway intersects a runway or another taxiway are seldom representative of the real world; triangular gussets, or none at all are more usual;
- taxiway centerlines:
 - are drawn towards all possible destinations when taxiways intersect, resulting in what I refer to as “spider-webs”; and
 - merge with the runway centerline at taxiway/runway intersections; at Canadian airports (and, I suspect, also those in some other countries), taxiway centerlines run alongside runway centerlines;
- taxiway edge lighting spacing can be very erratic when multiple links are involved, and
- if you attempt to use more than one taxiway/apron surface material, well ... we’ll get into that later.

This tutorial discusses a number of techniques for overcoming these issues and for otherwise adding realism to your MFS airport. As well, it suggests some methods for coping with the way MFS handles transitions from one surface material to another and other drawing-order issues.

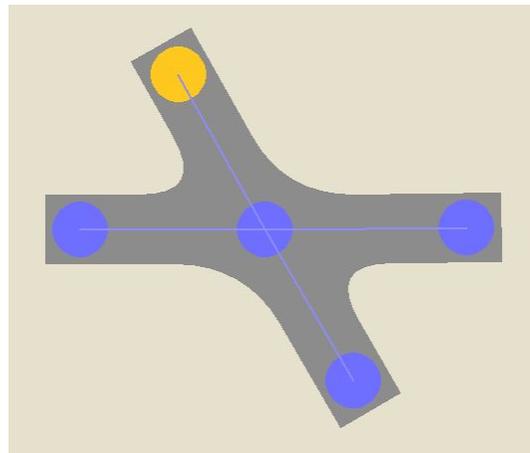
In writing this tutorial, I have assumed the reader is familiar with the terminology, concepts and fundamental elements involved in airport design for MFS.

Many of you will note that the illustrations in this tutorial were generated using the venerable AFCAD 2.21. While AFCAD 2.21 is not compatible with FSX (I favour ADE for FSX-work), for clarity, many of the illustrations needed a WYSIWYG editor. But, at time of writing, ADE was not yet WYSIWYG.

ELIMINATING FILLETS

When a taxiway intersects another taxiway or runway, MFS draws a rounded fillet at each “corner”. However, at real airports, especially large ones, corners which aircraft are intended to turn across, or corners which are filled-in to avoid aircraft accidentally travelling off the hard surface are more likely to have triangular gussets. On the other hand, there is often no filler at all when very wide turns are involved.

Unfortunately, Microsoft did not give us a mechanism to directly prevent these rounded fillets from being

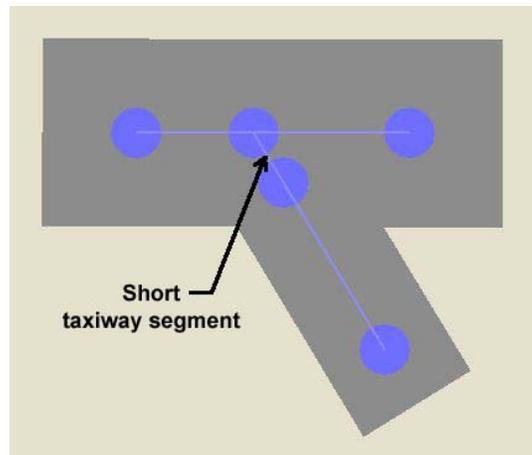


Default Fillets

drawn. But, there are several ways to suppress or hide them.

At the risk of stating the obvious, if there's no intersection; there's no fillet. Just because two taxiways cross, for example, or a taxiway joins a runway, doesn't mean the underlying links must be joined. On the MFS display, two unconnected crossing taxiways look just like their connected counterparts except that there are no fillets. Likewise, a taxiway "dead-ended" under a runway looks just like one that is connected to the runway link save, again that there are no fillets. But, you say, "How will AI aircraft turn at the intersection or exit the runway?" Simple! Add one or more connecting apron links, as necessary, to route the AI. Since apron links have no surface, they don't generate any fillets. You may observe as well that centerlines reflecting the turn are not possible. That's true. But, you'll find solutions for this little problem below. Bottom line, if you don't want fillets, don't connect links at "Ts" or other intersections unless there's a reason to.

If AI must use an intersection and you don't want to be bothered with extra apron links, adding an additional link to the taxiway very close to the intersection will also suppress the fillet. The size of a fillet on the MFS display is a function of the taxiway width. The fillet extends along the intersecting taxiways a distance of one taxiway width. Thus, wide taxiways have large fillets and narrow taxiways have small ones. But, one useful other characteristic is that if a taxiway extends beyond the intersecting taxiway (or runway) for a distance of less than one runway width, the corresponding fillet will be shortened accordingly. Taken to the extreme, if a taxiway extends only to the edge of the intersecting runway (or not even that far), the length of the fillet is reduced to 0. So, by placing an extra node in a taxiway under the intersecting taxiway or runway, the fillet is eliminated. But, be aware (for reasons discussed later) that the taxiway centerline will no longer be a sweeping curve. This can be fixed, however.



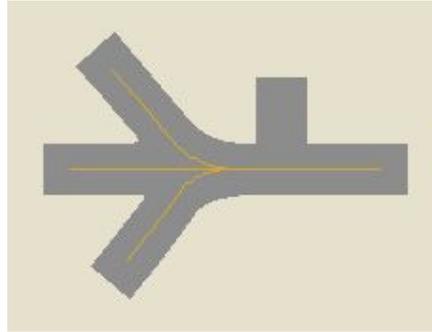
Fillets eliminated using a short, narrow taxiway segment

At taxiway/runway intersections, another possibility is to "hide" the fillet under the runway. While we think of such an intersection being between the runway and taxiway, in MFS the intersection is between the runway link and the taxiway link. Even though a runway may be 200' wide, the runway link can be narrower. If the width of the runway link is reduced to say 100', the fillet resulting from an intersection with a taxiway up to 100' wide will be totally hidden underneath the runway surface. However, once again, it may not be possible to have a centerline curve of the desired (large) radius.

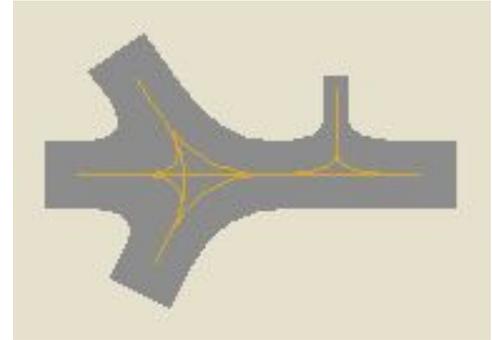
Finally, and simplest of all if it is applicable, simply place an apron over the fillet(s).

TAXIWAY CENTERLINES

Unless the airport designer takes special measures, taxiway centerline configurations are likely to be much more complex than desired. Consider, the illustrations to the right which depict a main taxiway leading to three aprons. The short stub leading up is a service road.



What you wanted



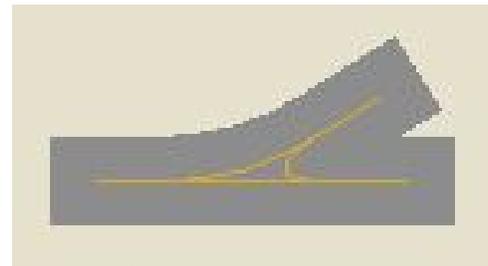
What you got

The first illustration is likely what you want as a centerline configuration. However, without special measures, what you get is shown in the other illustration.

When the MFS' drawing engine encounters an intersection of two links, one with centerline enabled, the other without, the centerline from the first link will be extended into the other for a distance of one link width. (For the moment, let's not worry about links of different widths.) If more than two links connect at the intersection, the centerline(s) of all the ones with centerline enabled extend for a similar distance into all the others. This has the potential to become a real mess and is the basis of the term "spiderweb" I've coined to apply to such situations. At first glance, this may seem irritating. But, in fact, it is the key to simplifying taxiway centerline configurations.

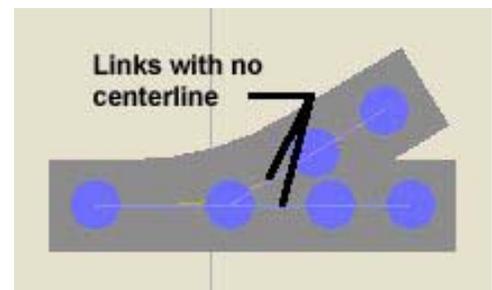
Consider the case where three intersecting links form a "Y". If two (or all three) links are designate as centerline-enabled, the result will be as shown to the right.

For purposes of illustration, let's say the short centerline segment between the two legs of the "Y" is not desired. We can't suppress the centerline extensions entirely. But, if we specify that only the link forming the main "leg" of the "Y" is to be-centerline enabled, we get the centerline from that leg extended into each "arm" without the intervening "short turn". To make the centerlines continue along both arms, we then add a centerline-enabled link at the end of each arm. Now, the "Y" is drawn as we want it.



Simple taxiway "Y"

The only time this centerline extension does not occur is into links having a width of 0 – and, of course, when none of the links involved are specified to have centerline enabled. Referring back to the "What you wanted/What you got" illustrations above, the centerline of the service road can be completely suppressed by connecting it to the main taxiway with a short 0-width link that will be hidden beneath the main taxiway.



Centerline segment suppressed

This logic can be extended to draw very complicated taxiway centerline configurations with only the desired centerline segments being drawn. However, there will be situations where drawing the desired centerline configuration becomes too complex, in which case you may wish to resort to decorative elements, covered below.

The radius of curvature of the centerline of a link is directly related to the width of the link. The wider the link, the greater the radius of curvature of its centerline. Given two long links which lie at an angle to each other, MFS starts and ends the curve in the centerline one link width on either side of the node at which the two links intersect. Extended curves, i.e., centerlines of curved taxiways made up of several links, are actually a series of individual curves end-to-end. Each individual curve begins and ends halfway to the adjacent node, to a maximum of one link width. Thus, the “perfect” extended curve is made up of a series of uniform links not longer than two link-widths and lying at uniform angles to each other.

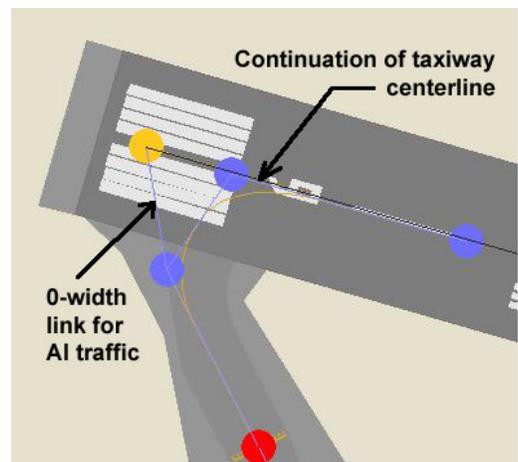
But, in the simple case of just two links, what happens if one of the links is shorter than twice its width? Like the size of filets, the radius of curvature of the centerline is also limited by link length. As a link becomes shorter, the radius of curvature of its centerline is correspondingly reduced.

While you may not have much flexibility with the width of visible taxiway segments, generally you are not constrained in the width of TAXI-type links that are drawn over aprons or underneath runways. At entrances to runways, you will often have to draw an apron to reflect real life. Since taxiways and aprons of the same surface material use the same texture, the width of a TAXI-type link in an areas covered by an apron can often be adjusted as necessary for a smoothly curving centerline.

While the airport designer can exercise a good deal of influence over the shape and configuration of the centerlines of TAXI-type links, there will be times when that’s not enough. As well, doing what you want solely by controlling TAXI-type link length and selectively enabling centerlines can lead to overly complex solutions. Fortunately, there’s another alternative.

Centerlines in MFS are for decorative purposes only. While it would be nice if they did, AI aircraft travel from node-to-node in straight lines. They do not follow the taxiway centerline. So, sometimes the easiest way to get the desired centerline configuration is to draw the taxiways without centerlines and carefully overlay them with (surface-less) PATH-type links where you will have complete freedom to specify taxiway width and, hence, the radius of curvature of their centerlines.

At large airports in Canada, and presumably also elsewhere, taxiway centerlines that extend onto the runway do not join the runway centerline. Rather, the taxiway centerline lays along side the runway centerline. However, for AI to work, the taxiway must be connected to the runway link (which usually lies along the centerline of the runway). This dilemma is easily resolved. Extend the taxiway centerline onto the runway



Decorative Centerline

using PATH-type links but connect the taxiway to the runway link using a 0-width link (to suppress the centerline)

OTHER DECORATIVE ELEMENTS

Airports have lots of surface lines. Depending on their color, these lines often can be drawn using the center, edge-lines and/or the surface of links. Taxiway edge lines are great for edging aprons as well, or for any place double yellow lines (dashed or solid) are required (Make the link width quite small (5' or less) so that the edge line will follow the edge of the apron.) Link centerlines can be used to draw a single yellow line. For white lines, use one or a series of narrow links with ICE or SNOW as the surface texture.

Such decorative lines will usually trigger errors in airport editor fault-finders as orphan or isolated links. To avoid many of these errors and to differentiate decorative links from those over which AI may travel, I use CLOSED-type links for decorative purposes whenever possible. (ADE's fault-finder and the fault-finders of some other editors ignore CLOSED-type links. As well, CLOSED-type links are usually shown in a different color on the editor's display. MFS displays CLOSED-type links as normal TAXI-type.) Errors that cannot be avoided in this way can, in normal circumstances, be ignored, but it is wise to ensure unexpected difficulties will not result.

For the MFS AI/ATC to perform as intended, a hold-short node is required at every runway entrance and exit. But the hold-short line associated with these nodes is purely for decorative purposes. These lines often are of a configuration not easily accomplished with a hold-short node in the MFS taxiway network (e.g., very long lines or at odd angles). Many airports also have hold-short (and ILS sold short) lines in places where hold-short nodes are not required by the MFS ATC engine. Decorative hold-short lines to the rescue!

To display a hold-short line, all that is required is a hold-short node connected to a normal node by a link of the correct width and orientation. Using PATH-type links for that purpose, hold-short lines may be placed anywhere on your airport – and not interfere with ATC/AI operation. Indeed, where a required hold short line is not easily drawn in the taxiway network, an alternative is to place an invisible hold-short node at the proper place in the taxiway network and then place a decorative hold short line on top of it.

However, hold-short nodes in the taxiway network are connected to two or more links. The displayed width and orientation of the associated hold-short line is derived from only one of those links. But, there is no requirement that the characteristics of these links be identical. So, this begs the question “Which link?”

For any hold-short node, the last link encountered by MFS that is connected to the hold-short node is the one that defines the width and orientation of the hold-short line. This will generally be the last entered of the connected links. However, if your airport editor changes the relative position of links (as, for example, ADE does when link orientation is reversed), the width and/or orientation of hold-short nodes can change unexpectedly. So, after editing an airport, you should confirm that your hold-short nodes have not inadvertently been affected.

Where aprons are used extensively to “decorate the taxiway network or where taxiways involve numerous short links, standard taxiway edge lighting is likely to be less than satisfactory. So, you may wish to avoid using taxiway lighting in such circumstances and, instead, line the edges of such taxiways with apron edge lights.

DRAWING ORDER MATTERS

As you probably have already discovered, with MFS, things are not always drawn as expected. Particularly insidious are apparently legitimate configurations of airport elements that appear normal in the display of whatever airport editor you may be using, but that are not displayed as intended by MFS. The problem often turns out to be drawing order issues.

Drawing order issues aren't likely to arise at airports that use a single surface texture and whose taxiways are comprised of the same type of links of uniform width. So, if that describes your airport, you can skip this section (which is going to get rather tedious). But, if your airport has:

- a contiguous series of links:
 - of non-uniform width or whose surface texture changes somewhere along its length,
 - of one surface material running over an apron of another, or
 - incorporating both TAXI-type and PATH-type links, or
- overlapping aprons of different surface materials,

perhaps you should read on. 'Cause chances are, things aren't gonna' work as you intend the first time.

The relative order in which airport elements are drawn by MFS has a major influence on what you see. To some extent, the order is governed by the order in which the elements are specified to MFS. That order is established in the XML file, which is generated by the airport editor and compiled using BGLComp (perhaps a custom version) into the airport BGL file input to MFS. (Any reference in this section to the relative order of airport elements - runways, links, taxiway signs, etc - refers to the position in the XML file of the element relative to other elements in the same collection – “earlier” implying closer to the beginning of the XML file and “later” closer to the end.) But, within collections of airport elements having surface texture, i.e, runways, links and aprons, the final drawing order is further dependent on the respective surface materials, with all elements having one surface material being drawn (in file order) before any element having another. Specific material priority is discussed later in this section.

You may have noticed from the above itemization that the noted issues all involve transitions – transitions from one surface material to the other, from one link width to another and from one link type to another. Fundamentally, it appears that the element drawn first by MFS overrides certain characteristics of the adjacent element(s) drawn later insofar as the drawing of the adjacent element is concerned. This is of no consequence where the critical parameters of the adjacent elements are identical. But at transitions, drawing order can have a major visible impact. Where there is a surface texture change, that change may occur one link either side of the intended node. Where a PATH-type link is joined to a TAXI-type link, the surface of the TAXI link may disappear or the PATH link may take on the surface of its neighbour, again depending on the specified surface material. Where there are overlapping aprons of differing surface materials, the wrong one may be on top.

Where these situations involve elements of the same surface material, e.g., varying link widths, , you may be able to correct the situation by forcing to the bottom of the XML file the link whose characteristics you do not want – so it will be drawn last). In the case of links, whether or not the direction in which the link is specified (start and end nodes) is consequential in such cases has not been confirmed, but is thought not to be a factor.

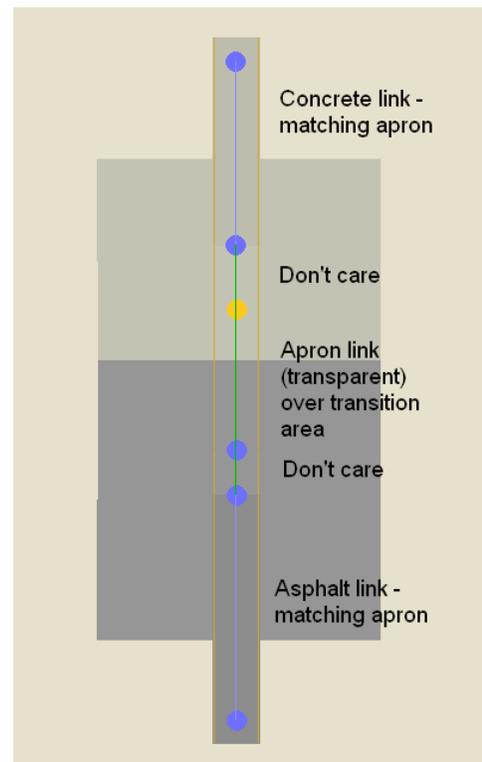
(The amount of control you have over drawing order depends on the airport editing tool you are using. Generally, airport design/editing tools place a newly added element last in its collection. Consequently, to change the relative position of an element with respect to the other elements in the same collection, you can delete and re-enter the one you want to be drawn last. Some tools, for example ADE, allow you to directly change the relative position of an element as well as the forward/reverse orientation of links. As a last resort, you can directly modify the XML and reload it or the compiled BGL back into the editor.)

Overlapping apron issues are generally easily handled once the material priority is taken into account. But, to address drawing order issues involving other transitions from one surface material to another, it is necessary to use what I call “don’t care” links. A “don’t care” link is a short, sacrificial link placed in the vicinity of a surface material transition to “absorb the effects” of the drawing order issue.

At simple taxiway surface transitions, place a “don’t care” TAXI-type link having as its surface the latter-drawn material at the desired transition point to the side of the material drawn first. If there is a drawing-order issue, that link will be surfaced with the first-drawn material instead. However, since it is specified to be surfaced with the other material, the transition effects will end there and the visible taxiway surface transition will occur at the desired point.

For a taxiway passing over an apron surface material transition, unless everything is perfectly aligned, a PATH-type (or apron) link must cover the apron transition area. But this constitutes two additional transitions, i.e, from the TAXI-type link of the first surface material, to the PATH-type link and then to the TAXI-type link of the second material. In such cases, two “don’t care” PATH-type links must be added, one at each end of PATH-type link covering the transition in the apron materials as shown in the illustration to the right.

In each case, the “don’t care” links must be added after the other link elements are in place to ensure that they are drawn after the other links and, hence, “absorb” the transition effects.



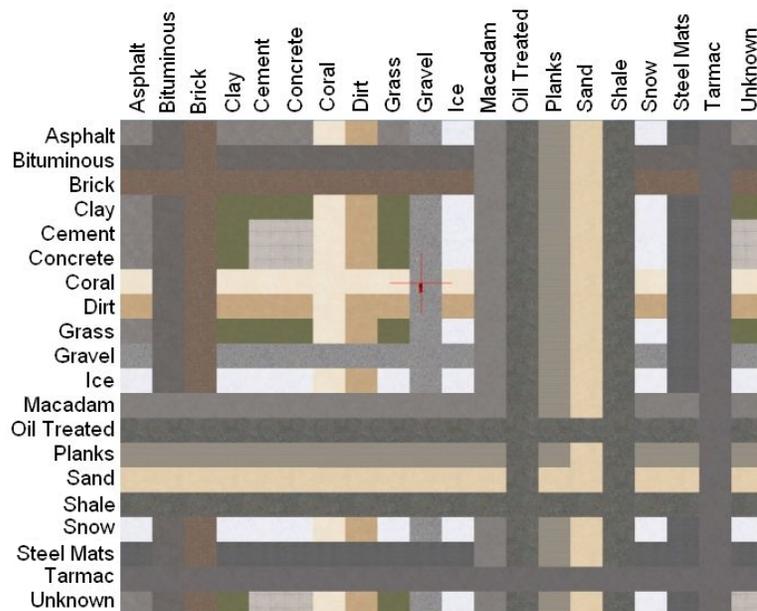
Apron transition

Generally, if you encounter a situation in the area of a transition of some type for which there is no plausible explanation based on BGLComp “rules”, suspect drawing order issues. Then, if you think about what’s happening, the solution shouldn’t be too evasive.

Taxiway and Apron Surface Material Priority - Runways, taxiways and aprons are drawn in a priority determined not only by their order of appearance within the XML file, but also by their surface material. Within each category, elements having as their surface the lowest priority material are drawn first and the highest priority material is drawn last. Hence, an apron of higher priority material will appear on top of one with a lower priority material.

Surfaces are drawn in the following order:

- CEMENT and CONCRETE (these surfaces appear identical)
- CLAY
- ASPHALT
- GRASS
- ICE and SNOW (these surfaces appear identical)
- DIRT
- CORAL
- GRAVEL
- STEEL_MATS
- BITUMINOUS
- BRICK
- MACADAM
- PLANKS
- SAND
- SHALE
- OIL_TREATED
- SHALE
- TARMAC



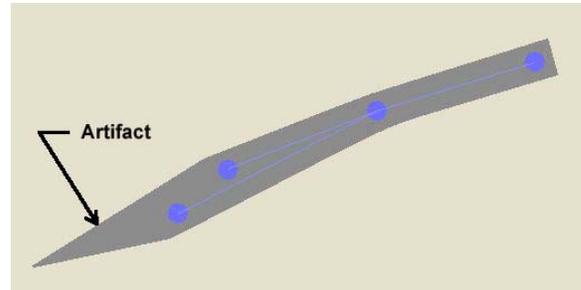
So, for example, on an ASPHALT apron with a CONCRETE pad, the pad outline should be drawn in the ASPHALT apron. The shape of the CONCRETE apron need only be confined to within the bounds of the other and cover the pad area.

MFS DOESNT LIKE SMALL ANGLES

There are (at least) three situations where MFS mishandles small angular differences between TAXI-type links.

- If two or more links connected end-to-end have the same, or almost the same, orientation, MFS sometimes seems to ignore the characteristics of all but the first link in the series - presumably the first drawn. This has little or no visual effect if all links are otherwise identical. However, if they’re not, you can “scratch your head” a long while before you realize what’s happening. If this is the situation, moving one or more of the nodes to put a slight, unnoticeable “kink” in the line can have a significant effect on the display of the involved links. This may be another manifestation of a drawing-order issue.

- If you place three links in a “Y” configuration with only a very small angle between the “arms”, you may find an artifact emanating from the junction or intersection. This artifact only seems to occur at specific angles and can usually be eliminated by slightly adjusting the angle of the intersection.
- A similar artifact is sometimes found with a series of three (or more) links connected end-to-end where one link makes a small angle with the others. As above, such artifacts can usually be overcome by a slight adjustment of the angles involved.



Artifact from Very Acute Angle

Depending on the algorithms used, these artifacts may or may not appear in the display of your airport editor. As well, your airport editor may show artifacts where none appear in the MFS display. So, finding and getting rid of such artifacts is often an exercise of trial and error.

FINAL ROUNDUP

Nodes Don't Come for Free – I have discussed rather glibly adding nodes and links to overcome various situations or to add realism. However there is a cost in doing so. First is the intuitive, FPS “hit”. The more nodes to be processed, the more processing time required and, hence, the lower the FPS. However, there are other aspects of airport scenery that seem to have a far greater impact on FPS. A reasonable number of extra nodes/links is unlikely to have a noticeable impact FPS.

More apparent, however, is the “hit” on AI taxi speed. MFS performs significant calculations in respect of an AI aircraft at every node along its taxi path. What is being calculated is unimportant - since there's nothing we can do about it. But, where an AI aircraft encounters several nodes in a relatively short distance, there may be a noticeable slowing of its taxi speed in that area. Consequently, where several nodes in close proximity cause such a slowdown, if these nodes are not otherwise relevant to AI performance, you may wish to consider removing them from paths used by AI and re-implementing the feature by means of decorative elements.

0-Width Links – A 0-width link provides a path that AI will follow but that has no surface, sidelines or centerline. 0-width links are useful for eliminating/managing spider-webs and for drawing invisible hold-short nodes.

But, use of 0-width links should be minimized. According to Holger Sandmann, a well-known scenery design guru, apron routes must have a minimum width of 20' (6 m.) for proper collision avoidance in FS9. As well, various forums contain warnings about using 0-width links. Nonetheless, I have not observed any ill-effects using 0-width links to the limited extent described in this tutorial.

Good luck with your airport,
 Don Grovestine
cyj2007@shaw.ca
 May 1, 2008