“DRIVE_THROUGH” PARKING
and OTHER AI AIRCRAFT CONSIDERATIONS

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

One of the major tasks in developing CYYJ (2007) – a FS9 and FSX compatible add-on covering the Victoria (BC, Canada) International Airport and other nearby facilities – was the implementation of “drive-through” (sometimes called “forward-exit”) parking, which is used exclusively at the real CYYJ. My focus while doing so was on realism and, I gather as a consequence, I must have attempted some unusual (though legitimate) techniques. Because, at every turn, I seemed to encounter another obstacle thrown-up by MFS. Few of these situations pertained exclusively to “drive-through” parking. Most were generic AI problems. I spent considerable time trying to understand what was going on “inside” MFS, in order to find a way to overcome the problems. I learned of some unusual things about MFS’ processing of AI, things that (based on the reaction to several posts I made to various forums) apparently are not well known. I thought it useful to collect all this new-found knowledge in one place. Hence, this tutorial.

This tutorial does not include any illustrations. While many of the concepts involved are simple, their implementation is generally quite complex – too complex for simple illustrations. Therefore, I suggest you refer to the airport definition file from either the FS9 (AF2_CYYJ.bgl) or FSX (AFX_CYYJ.bgl or ADE_CYYJ.bgl) version of CYYJ (2007), which are all equivalent for the purposes of this tutorial – using whatever airport editor you’re comfortable with. I should also point out that the two-network version of CYYJ (2007) referred to herein has not yet been released and likely won’t be for at least a few weeks. However, all the principles outlined in this tutorial apply equally to the currently-released three-network version

Incidentally, this is the second edition of this tutorial. Aside from generally updating its predecessor to include FSX, it also contains additional information of some of the MFS “gotchas”.

BACKGROUND

The Help feature of Lee Swordy’s AFCAD 2.21, the standard tool for FS9 airport development, states “because of limitations with the AI, it [drive-through parking] rarely results in the kind of behaviour one would hope”. However, from correspondence with Jon Patch, I discovered he and Holger Sandmann, both well-known freeware- and payware-scenery developers, had successfully implemented “drive-through” parking at their CYWH water airport. A series of posts by Jon and Holger in the AVSIM forum outlining the key factors in their implementation appears at http://forums.avsim.net/dcboard.php?az=show_mesg&forum=123&topic_id=28566&mesg_id=28566&listing_type=search.

In summary, those factors are:

- separate paths must be provided for arriving and departing AI;

- the departure path must connect the outbound side of the parking spots only to the nodes closest to the ends of the runways through hold-short nodes within the prescribed maximum distance (225’) of the edge of the runway; and
- the arrivals path must connect the inbound sides of the parking spots to all intermediate points on the runway where traffic is to exit.

This general approach is commonly referred to as “plumbing” – due to the use of parallel networks of links that, in plan view, suggest the hot- and cold-water plumbing in a house. I later discovered that Jon and Holger were not the only ones to use this technique and, in fact, at least two tutorials had been written on the topic. (However, it would seem the earlier tutorials are either entirely theoretical or their authors were extremely lucky in their implementations.)

I examined Jon’s and Holger’s CYWH implementation. Since it was a water airport, there was no defined taxiway structure. They could route traffic wherever they wanted. Also, conveniently, all AI parked in one general area. I wanted to apply their principles to a land-based airport with three runways, eleven taxiways and several aircraft parking areas.

(CYYJ was my first major scenery design project and unfortunately, due to my lack of experience, “drive-through” parking considerations were left until after the runway/taxiway/parking networks were complete. But, I’m sure others will attempt to add “drive-through” parking to an existing (stock) airport. Hence, I felt my experiences in this regard may still be relevant. If you’re not in that situation, you may wish to skip ahead to the next section.)

Initially, using PATH-type links, I overlaid the existing runway/taxiways with two invisible taxiway networks – one for departures and another for arrivals. AI traffic was to use these taxiways, leaving the visible taxiway network (which I’ll call the “base network”) completely passive insofar as AI is concerned. While I felt it should be possible to use the taxiways in the base network for arriving traffic, which I later confirmed, I wanted to keep my first attempt at “drive-through” parking as straightforward as possible. To my pleasant surprise, the three-network (arrivals, departure and base) solution worked pretty much as I had intended – once I addressed the MFS “gotchas”.

Since I could find nothing in the forums regarding these difficulties and since there was no mention of them in the existing tutorials, I posted several lengthy accounts of my experiences in the hopes of helping others avoid them. Those posts met a variety of reactions – from gratitude to derision. (The problems I had encountered didn’t really exist according to some). But, a common reaction also was “Why three networks?” when two would do.

Theoretically, two networks should do. So, I re-implemented my scheme - dispensing with the separate arrivals network – instead using the base network for arrivals. While I was successful, I found it very difficult to integrate the arrivals network into a base network – the latter of which had taken on additional complexity due to my having addressed some other realism issues (see the companion tutorial “Adding to the Realism of your Airport”). The combined base/arrivals network became so complex that I reverted to the three-network configuration – which is still the currently released configuration.

However, it continued to frustrate me that I'd not been very successful in eliminating the third (arrivals) network at CYYJ. So, some time later, I decided to give it another try. With my new-found knowledge (and a lot more experience), I undertook what was essentially a complete redesign of the whole airport. In doing so, I significantly simplified my CYYJ "afcad". ("afcad" is in parenthesis here because the redesign was undertaken using Jon Masterson's ADE.) This two-network version will be released along with other improvement to CYYJ (2007) in mid-2008.

"GOTCHAS" AND OTHER MFS AI/ATC ENGINE FUNDAMENTALS

Hold-Short Nodes – The MFS AI/ATC engine requires hold short nodes be placed to cover two situations:

- each departing AI must encounter a hold-short node before entering the destination runway; and
- each arriving AI must encounter a hold-short node before reaching its parking spot.

A single node may serve both purposes.

Hold-short nodes need not terminate the link leading directly from a RUNWAY-type link. There may be several intervening links. The only firm positioning requirement is that hold-short nodes for departures lie within 225' (~70m.) of some point of the surface of the runway to which they apply. Hold-short nodes for arrivals may be anywhere in the taxiway network (provided, of course, AI cross them before arriving at the parking area). But, since AI are issued parking instructions at the first hold-short node encountered after leaving the runway, these nodes should be located near where AI enter the taxiway network.

In "drive-through" parking applications, there must be two hold-short nodes near the ends of the runways, one in the arrivals plumbing and one for departures. If arrivals use the visible taxiway network, the hold-short node associated with the visible hold-short line will serve that purpose. If not, an invisible hold-short node in the arrivals network will be required in addition to the invisible node in the departures network.

While FSX provides for invisible hold-short nodes, FS9 does not. But, in either version, a hold-short line will not be displayed if the width of the link controlling the operation of the hold-short node is 1' or less – the controlling link being the last-drawn link connected to the hold-short node. So, to create an invisible hold-short node in an FS9 AFCAD or in an FSX airport definition that is to also be used for FS9, place the hold short node where needed in the taxi network and replace the link on the runway side of the hold-short node with another of 0 or 1' width. (Replacing the link results in a new link being placed at the end of the collection of links and, hence, becoming the one that controls the display of the hold-short line. While replacing either connected link with one of 0-width would suppress the display of the hold-short line, the runway-side link should be the one replaced. According to Holger Sandmann, the taxiway or apron route leading to a "departure" hold-short node must have a minimum width of 20' (6 m.) to avoid AI "piling –up" at the hold-short node.)

Taxi-to-Parking Clearance – Immediately after an AI aircraft lands, MFS determines whether suitable parking is available for it and, if so, reserves a particular parking spot and, apparently, determines the route be taken to parking. If no suitable parking is available, the AI will disappear while still on the runway.

Once clear of the runway, arriving AI travel towards the reserved parking spot until they encounter a hold-short node. Immediately upon crossing this hold-short node, the AI stop to seek clearance to taxi-to-parking.. At this point, it appears MFS recalculates the route to parking for ATC purposes, commencing with the node to which the AI is then closest. If no route exists between the closest node and the reserved parking spot, once again, the AI will simply disappear.

When an AI stops to receive ATC taxi-to-parking clearance, it is unlikely to be sitting over a node. Since the AI stops after passing over the hold-short node, the hold-short node is behind it. The position in which the AI stops is governed by the size of the aircraft. Different AI aircraft will, therefore, stop in slightly different places. AI travels from node to node; MFS selects the the closest node beyond the hold-short node as the starting point for the journey to parking. Once ATC taxi-to-parking clearance has been issued, the AI proceeds to the closest node and onwards to parking.

It doesn't matter whether the closest node is part of the taxiway network, a decorative element or, in drive-through parking applications, in the departure plumbing, The sole criteria appears to be "closest". As a consequence, depending on node configuration, an arriving aircraft could jump to the departure plumbing, another taxiway or even to a decorative runway edge line that leads nowhere.

To ensure the AI remains on the arrivals network, the area on the side away from the runway of hold short nodes used by arriving AI should be kept clear of all nodes other than those in the arrivals network. If the area cannot be kept clear of other nodes, sufficient additional arrivals nodes must be placed in the area to ensure that any AI aircraft stopping at that hold-short node to receive clearance will find itself closest to an arrivals node. As well, if there are no nearby suitable nodes in the arrivals network, you should place one in the arrivals path just beyond that area such that it will be the node closest to any AI awaiting clearance. As part of this strategy, any invisible hold-short nodes in the departure plumbing should be placed slightly (but more that the critical distance noted below) to the runway side of the arrivals hold-short node. If you insist on placing a departure hold-short node on the parking side of an the arrivals hold-short node (which is where it intuitively should be), you MUST place an extra regular node in the arrivals plumbing next to it to avoid the departure hold-short node "capturing" arriving AI.)

While on the topic of AI travelling from node to node, don't expect AI to travel along those smooth, wide-radius turns of your taxiway centerlines. AI travels in a straight line from one node to the next. When AI does turn, the radius of the turn depends not on the radius of the taxiway curve but, rather, on the nose-wheel maximum angle/turning radius specified for the AI aircraft in its aircraft.cfg file. Consequently, if you want your AI to make wide, sweeping turns, you're going to have to add additional links/nodes where the taxiway curves.

Other Node Proximity Difficulties - If two links terminate on nodes very close to one another, all may appear OK on the airport editor display. However MFS may not draw or route AI along the associated links as you intend.

When drawing a link, MFS appears first to check whether there are any other nodes within a critical distance, about 8' or 2.5 m., of the starting node. If MFS encounters an older node (i.e., one entered earlier than the node that starts the link of interest) within that distance, the link simply is not drawn. If there are no older nodes, a link will be drawn – but not necessarily to the intended destination. Before drawing the link, MFS also appears to check whether or not there are any other nodes within the same critical distance of the designated destination node. If not, the link will be drawn as intended. But, if there are other nodes within that range, MFS will draw the link from the specified starting node to the node-in-range which is furthest away from the starting node – irrespective of where you told it to draw. None of this is will be evident from the airport editor display and, because of the very short distances involved, likely won't be noticeable on the MFS display either. But, such situations can result in "broken" taxi paths and interconnection of arrival and departure "plumbing" in drive-through applications. If either happens, it's unlikely the AI at your airport will work as expected.

The most common symptoms of these node-proximity difficulties are AI disappearing immediately after receiving taxi-to-parking clearance even though you appear to have provided a proper path for them or, in drive-through applications, arriving AI mysteriously finding their way to the departures plumbing and vice versa. If you experience such problems, enabling the centerlines of the arrivals and departure plumbing and checking the MFS display may help confirm the situation.

Due to the limited number of such situations encountered, the details above of what I think MFS does when it encounters nodes in close proximity may not be exactly correct. Perhaps it's not the older node or the one furthest away. But, one thing is for certain. When nodes are in very close proximity to one another, MFS may not do what was intended.

Parking Connector Width/Radius of Turn – For AI to depart properly, it is essential they park close to the center of the parking spot. (AI spawned in place are not of concern, since they will be positioned exactly on the parking spot.) It has been estimated that AI requires approximately two aircraft lengths to properly align itself with the parking spot after a sharp turn. If the entry to a parking spot is tightly-curved and sufficient straight lead-in is not provided, the AI may be some distance from the center of and/or be incorrectly oriented with the parking spot when it stops.

When initially developing CYYJ (2007) for FS9, I experienced some difficulties with badly-parked aircraft pushing-back rather than proceeding forward out of their parking spots. In retrospect, there were several possible causes for that behaviour. However, with FSX, arriving AI that parked too far from the center of parking spots simply refuse to depart. I observed his behaviour primarily from AI parked in parking spots that had a sharply-curved arrival path.

It seems that early in the departure sequence, FSX tests whether an AI is validly parked by confirming it to be within a certain distance of the center of a parking spot. That test distance appears to be the width of the parking connectors. (An AI parked beyond this test distance does not progress further through the departure sequence. I found that, at CYYJ, I needed a minimum parking connector width of about 15' (5m.) on sharply curved entries with limited lead-in to ensure AI departed normally.

Avoiding Pushback – In FSX, departing AI always push back, irrespective of the "Pushback" specification in the XML file. The length of the pushback is the shorter of:

- the radius of the parking spot, and
- the distance from the center of the parking spot to the closest node.

In order to avoid noticeable pushback with drive-through parking, I found it necessary to place the first node in the departures plumbing very close to the parking spot center. As well, to avoid (or at least minimize) a "twitch" by the AI as it prepares to depart, that node must be in line with the parking spot orientation.

I also found that in both FS9 and FSX, AI dislikes making turns greater than 90° from a parking spot. Consequently, where an AI must make a sharp turn upon leaving a parking spot, I place the second node in the departure plumbing slightly beyond the first to the side I wish the AI to turn, limiting the initial turn to 60-80°. If the desired total turn approaches or is >180°, the second node must be placed far to the correct side or a third node used to split the turn.

AI Will Transit "Black" (RUNWAY-type) Links – Common knowledge has it that MFS will not route AI over a "black" (RUNWAY-type) link if another route to the destination is available. This is one of the foundations of the "plumbing" method for drive-through parking. But, it is not true, or at least it is not true with departing FS9 AI. MFS will route departing FS9 AI over the shortest route irrespective of whether or not that route contains a "black" link. (I have never observed this behavior with arriving FS9 AI or with any FSX AI.)

To make matters worse, in FS9 (but not FSX), the calculation of the length of the routes appears not to include the parking connectors (i.e., the last link into a parking spot and the first link out). Hence, in a drive-through parking situation with parking oriented away from the main taxiway, which likely would involve a long straight parking connector on the arrivals side, the shortest route by far from parking to the end of the runway is back along the arrivals path and then along the "black" (RUNWAY-type) link. To overcome this FS9 difficulty, place a node in the arrivals path just short of the center of the parking spot, thus forcing the arrival parking connector to be very short and the remainder of the length of the former parking connector to be included in the calculation of the relevant distances. This additional node should not be used except in drive-through parking applications, since it will limit the length of the pushback.

While I've not observed FSX routing AI traffic over "black" links, I still take great care in laying out the departure plumbing for both FS9 and FSX to ensure that the departure path to the end of any runway is shorter than any arrivals path plus the distance along the runway to the relevant end.

IMPLEMENTING “DRIVE-THROUGH” PARKING

Build your Drive-Through Airport – I have found the following sequence of operations to be a satisfactory way to create an airport which has drive-through parking.

- Lay out the runways and taxiways, including decorative aspects, without regard to how AI will operate. (Don’t worry about parking details at this time, but do consider how you’re going to get AI into and out or the major parking areas.) This step should be as complete as possible before proceeding, since any change in it later will likely affect what’s been done in all subsequent steps.
- Add the departure plumbing, that is, the routes from parking to the ends (but only the ends) of the runways. Departure plumbing must terminate on the end-most nodes of the series of runway links (the runway “spline”). (All the runway link nodes to which the arrivals plumbing is connected must be inboard of the departure nodes.) The primary consideration in laying out departure plumbing is that it be as short as possible. Cut across corners where you can. (Little bits here and there can add up to significant savings!) And be consistent in link placement. I’ve found the best approach is that departure plumbing always be on the side of the arrivals plumbing towards the center of the airport – except in the parking areas where it should be on the parking side of the arrivals plumbing. This would be a good time to take a look at the CYYJ layout if you haven’t already done so.
- Lay out the parking spots and associated links, but do not connect to the arrivals and departure plumbing until all those links are in their notionally-final position.
- Connect the parking-related links to the arrivals and departure plumbing. For arrivals, connect with a single link as close to perpendicular to the departures plumbing/taxiway as it can be made - thus maximizing the length of the associated arrivals plumbing. In order to make the departure paths as short as possible, the link(s) connecting the parking-related links to the departure plumbing should be:
    - where AI can turn onto the main taxiway in either direction, a Y-configuration with the “arms” extending as far to each side of the associated arrivals link and back towards the parking spot; or
    - where AI turns in one direction only, as long and at as shallow an angle;

  as is possible, consistent with airport realism.
- Test the AI. Create a traffic file with sufficient and appropriate aircraft to fill all the available parking in the parking area under test, scheduled to arrive at intervals of three- to four minutes, and all scheduled to depart a short time later. Run the simulator at 4x normal speed (the highest speed that will manipulate AI). Set the simulator time near the time that the last aircraft is due to depart. Now watch as each AI leaves its parking spot to ensure it departs in the forward direction and turns the right way. If the departing AI have a choice of which way to turn onto the main taxiway, repeat after first setting the wind direction to force runway selection such that the AI turn the other way. Once all the AI depart properly when spawned in place, set the simulator time such that the AI will arrive in sequence. Check that each AI actually arrives at its parking spot using appropriate routing and parks in the right direction. Listen to/watch the ATC taxi-to-parking clearances to ensure the paths are as intended and are adhered to by the AI. (This is where you’ll discover interconnected arrivals and departure networks.) Allow all

aircraft to arrive and depart. Repeat for each runway. When it all works, you've got your drive-through airport.

Finally, a word about conventions. While you may use any scheme you like, I use the following link types in the noted application:

- RUNWAY ("black") - runway splines,
- TAXI ("blue") - all physical taxiways and other arrivals plumbing,
- PATH ("green") - departure plumbing, parking connectors and decorative elements that cannot have a surface, and
- CLOSED ("red") - closed taxiways, taxiways that will not be use by AI and all decorative elements where a surfaced taxiway is appropriate; this excludes the links associated with decorative hold-short node which, almost always will have to be surface-less PATH-type links.

(The colors you will probably recognize as the default colors used by AFCAD 2.21.)

<u>Controlling Which Runway Exit is Used</u> - Purely by accident, I discovered that, while arriving AI generally use the first runway exit encountered after run-out, at times MFS can be "choosey".

FS9 appears to have an aversion to runway exits where the aircraft must turn through more than 90º. Indeed, AI will pass up such an exit to go further down the runway to reach one which has a lesser angle. I have not done any significant experimentation in this regard. Rather, I mention it here simply to make you aware of the possibilities. For example, if you have a situation where you want the AI from one direction not to use an exit, placing the link between the taxiway and the runway so as to increase the angle of intersection from that direction may force AI to bypass it. Of course, this may enable exits by traffic from the other direction, so how far down the runway the exit is located will also be a factor in how you design your runway exit strategy.

It also appears that MFS will only tolerate AI "running out" a limited distance before it will force the AI off the runway. If an AI has not found an exit off the runway link within this critical distance, it may "cut across the grass" to the nearest node. Consequently, with runways having exits at one end only, it may be necessary to insert additional nodes in the runway link connected to PATH- or TAXI-type links that lead back along the runway (to parking), thus allowing the AI to turn-around and taxi back to the nearest exit.

<u>Special Taxi Routes</u> - If the airport you are modeling has different taxi routes for different classes of aircraft, you may also be able to implement this using secondary plumbing.

At CYYJ, a hanger is located quite close to one of the taxiways, so close that it's not safe for wide-bodied aircraft to use that taxiway. Instead, wide bodied aircraft bypass this narrow taxiway by traveling part way to the terminal along a lightly-used runway. To replicate this, I created additional plumbing for wide-body arrivals and departures, extending from the point where the "wide-bodies" exit the taxiway in favor of the runway, along the runway, and then (only) to the parking spots for wide-bodied aircraft at the terminal (which are not connected to the main arrival and departure plumbing). Consequently, the wide-bodies' only paths between their parking and the point where they diverge/converge with the main taxiway network is this secondary plumbing.

A FINAL THOUGHT

While plumbing appears to offer a general solution for implementing "drive-through" parking, it is not the only solution.

MFS routes AI via the shortest available route to its destination. It doesn't just count links, it actually calculates each candidate-link length and, hence, the actual total length of each prospective taxi path – picking the shortest total path. Some have implemented a limited drive-through parking arrangement (e.g., for a specific parking area) based on this shortest-route logic. However, if arriving and departing AI share any portion of a taxiway, every taxiway that eventually links to that portion becomes a potential path for both arriving and departing AI. So, it's unlikely you will be able successfully to implement a "drive-through" parking scheme using shortest-route logic that will work across the whole airport with any runway being active.

But, where you can find a way to:

- segregate arriving and departing AI (in the vicinity of a take-off-only or landing-only runway, any given taxiway will carry either arriving or departing traffic, but not both; as well, by omitting strategic taxiway links such that AI is prevented from using certain portions of a taxiway, other parts of the taxiway may be made effectively one-way); or
- in a localized area, guarantee that the path from parking to takeoff will be shorter than the path for arrivals to parking (as would be the case if parking was located very close to the take-off end of the active runway);

it may be possible to implement "drive-through" parking for a portion of your airport without using plumbing techniques. But, this may also require artificial constraints on, for example, the choice of active runway(s) and, hence, loss of realism – and will certainly call for a good deal of inventiveness on your part.

Revisiting Lee Swordy's comment in the Help feature of AFCAD 2.21 "because of limitations with the AI, it [drive-through parking] rarely results in the kind of behaviour one would hope", I have to wonder if maybe he was referring to various node proximity difficulties discussed in this tutorial. If you think about it, properly-implemented plumbing ensures the correct operation of "drive-through" parking because any taxiways accessible to arriving AI lead only to the inbound side of the parking spots and all routes from the outbound side of the parking spots lead only to AI take-off positions. It can't fail (so long as "node-proximity" and some of the other "gotchas" I've discussed doesn't "get you")!

Good luck with your airport,

Don Grovestine
cyyj2007@shaw.ca
May 1, 2008