FLIGHT PLAN GROUP

The Flight Plan Data Group variables control the navigation engine of the gps. They cover Flight Planning to En Route Navigation to Instrument Approaches.

In my opinion, the flight navigation capability provided by the Flight Plan Data Group variables is thorough and pretty impressive, especially considering the inexpensive price of the software. The Flight Plan Group is by far the largest Group within fs9gps, containing 99 variables.

I find that understanding and predicting the response of the Flight Plan variables requires documentation of significant detail, at least for my purposes, which explains the length of this chapter. Still, there are things not covered and still not understood.

Flight Planning

COMPONENTS OF THE FLIGHT PLAN

The Flight Plan components are a group of read-only variables that are set through use of FS9's Flight Planner, mid-flight Flight Plan filing using FS9's ATC capability, user editing of the Flight Plan.PLN file, or through third-party Flight Planners (I've never investigated any, but several are out there that look pretty good).

- **FlightPlanTitle (string) [Get]**

FlightPlanTitle is created from Departure Airport Ident and Destination airport Ident. "Departure Airport Ident to Destination Airport Ident"

- **FlightPlanDescription (string) [Get]**

FlightPlanDescription is created from Departure Airport Ident and Destination airport Ident. "Departure Airport Ident, Destination Airport Ident"

- **FlightPlanFlightPlanType (enum) [Get]**

FlightPlanFlightPlanType is a number defining the type of Flight Plan.

- 0 = NONE
- 1 = VFR
- 2 = IFR

❑ FlightPlanRouteType (enum) [Get]

FlightPlanRouteType is an enum representing basic routing type. See table below.

ATC ROUTE TYPE

Bit	Name and Type #	Bit	Name and Type #
0	DIRECT = 0	2	LOWALT = 2
1	VOR = 1	3	HIGHALT = 3

http://msdn.microsoft.com/en-us/library/cc526954.aspx#ATC_ROUTETYPE

❑ FlightPlanCruisingAltitude (feet) [Get]

Discussed toward at the end of this section.

❑ FlightPlanDepartureAirportIdent (string) [Get]

The 3 to 4 character Ident of the Departure Airport.

❑ FlightPlanDepartureLatitude
❑ FlightPlanDepartureLongitude (degrees, radians) [Get]

Latitude and Longitude of the starting position of the aircraft at the departure airport. This could be parked at a gate or on a runway, and these will represent different coordinates. It is fixed at the starting point and does not change as the aircraft taxis for takeoff. If FS9 Flight Planner is used to position the aircraft on the Active Runway, then as shown in the figure below, the aircraft will be placed approximately 175 feet beyond the physical end of the runway. Units are degrees (decimal format, not deg, min, sec) or radians.

❑ FlightPlanDepartureAltitude (feet) [Get]

Altitude (asl) of the departure location, or, starting waypoint of the flight plan.

❑ FlightPlanDepartureName (string) [Get]

Name of the Departure Airport.

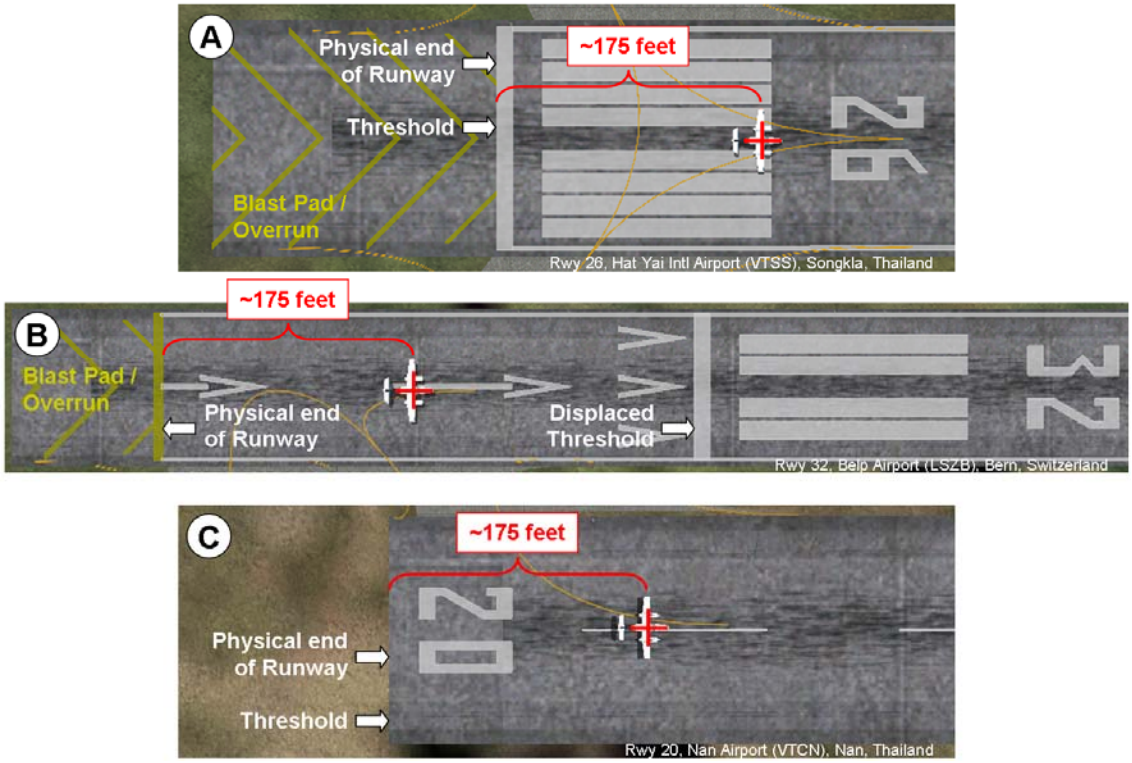
❑ FlightPlanDestinationAirportIdent (string) [Get]

The 3 to 4 character Destination airport Ident.

❑ **FlightPlanDestinationLatitude**
❑ **FlightPlanDestinationLongitude (degrees, radians) [Get]**

Latitude and Longitude of a "destination point" at the destination airport approximately 175 feet beyond the physical end of the approach end of the active runway, as shown below. Although landing is not permitted before a displaced threshold, it appears that for the sake of simplicity in the fs9gps world, the departure and destination points of the active runway are coincident, even in the case of displaced thresholds. Units are degrees (decimal format, not deg, min, sec) or radians.

FlightPlanDestinationLatitude, FlightPlanDestinationLongitude



Departure/Destination and Runway Approach Waypoints (△) are co-located with FlightPlanDestinationLatitude and Longitude at airports having an Instrument Approach Procedure.

❑ **FlightPlanDestinationAltitude (feet) [Get]**

Altitude (asl) of the destination point.

❑ **FlightPlanDestinationName (string) [Get]**

Destination airport Name.

The following table compares fs9gps variables to the equivalent entries of the Flight Plan .PLN file for a flight from Pochentong Airport (Phnom Penh International), Phnom Penh, Cambodia direct Changi Airport, Singapore.

fs9gps	FS9 Flight Plan IFR Pochentong Intl to Changi.PLN
	1. [flightplan]
	2. AppVersion=9.1.40901
FlightPlanTitle	3. title=VDPP to WSSS
FlightPlanDescription	4. description=VDPP, WSSS
FlightPlanFlightPlanType	5. type=IFR
FlightPlanRouteType	6. routetype=0
FlightPlanCruisingAltitude	7. cruising_altitude=34000
See id detail below	8. departure_id=VDPP, N11* 32.52', E104* 51.16', +000040.00
	9. departure_position=PARKING 1
See id detail below	10. destination_id=WSSS, N1* 19.75', E103* 59.11', +000021.99
FlightPlanDepartureName	11. departure_name=Pochentong Intl
FlightPlanDestinationName	12. destination_name=Changi
Flight Play Waypoint Info	13. waypoint.0=, VDPP, , VDPP, A, N11* 32.52', E104* 51.16', +000040.00,
	14. waypoint.1=, WSSS, , WSSS, A, N1* 19.75', E103* 59.11', +000021.99,

departure_id=VDPP, N11* 32.52', E104* 51.16', +000040.00

FlightPlanDepartureAirportIdent	VDPP
FlightPlanDepartureLatitude	N11* 32.52'
FlightPlanDepartureLongitude	E104* 51.16'
FlightPlanDepartureAltitude	+000040.00

And the associated fs9gps values:

```
GPS Viewer 1.2                  11111111112
GET SET SLEN 12345678901234567890 STRING NUMBER
 G       12  VDPP to WSSS         FlightPlanTitle, string
 G       10  VDPP, WSSS           FlightPlanDescription, string
 G        4  VDPP                 FlightPlanDepartureAirportIdent, string
 G       15  Pochentong Intl      FlightPlanDepartureName, string
 G        6     11.5420785        FlightPlanDepartureLatitude, degrees
 G        6    104.8526934        FlightPlanDepartureLongitude, degrees
 G        6   40.0000004          FlightPlanDepartureAltitude, feet
 G        4   WSSS                FlightPlanDestinationAirportIdent, string
 G        6   Changi              FlightPlanDestinationName, string
 G        6      1.3292220        FlightPlanDestinationLatitude, degrees
 G        6    103.9851643        FlightPlanDestinationLongitude, degrees
 G        5   21.998038           FlightPlanDestinationAltitude, feet
 G        0                       FlightPlanAlternateAirportIdent, string
 G        0                       FlightPlanAlternateName, string
 G        2      0.0000000        FlightPlanAlternateLatitude, degrees
 G        2      0.0000000        FlightPlanAlternateLongitude, degrees
 G        2   0.000000            FlightPlanAlternateAltitude, feet
```

- **FlightPlanAlternateAirportIdent (string) [Get]**
- **FlightPlanAlternateLatitude (degrees) [Get]**
- **FlightPlanAlternateLongitude (degrees) [Get]**
- **FlightPlanAlternateAltitude (feet) [Get]**
- **FlightPlanAlternateName (string) [Get]**

With the FlightPlanAlternate variables, an alternate airport or waypoint destination can be identified in the Flight Plan file. FS9 Flight Planner lacks the capability to create the Alternate, and the fs9gps FlightPlanAlternate variables are Get only, so the user must hand edit the .PLN file or use a third-party Flight Planner (if one exists that does this, I don't know) to define the Alternate.

I've added lines 13 and 14 to the .PLN file to define the Alternate Airport. For Changi, it is Hang Nadim Airport (Ident = WIKB), Batam, Indonesia, as shown below. I must obtain the airport Ident, Latitude, Longitude, Altitude and Name independently beforehand, and add it to the .PLN file.

fs9gps		**FS9 Flight Plan**
		IFR Pochentong Intl to Changi.PLN
	1.	[flightplan]
	2.	AppVersion=9.1.40901
FlightPlanTitle	3.	title=VDPP to WSSS
FlightPlanDescription	4.	description=VDPP, WSSS
FlightPlanFlightPlanType	5.	type=IFR
FlightPlanRouteType	6.	routetype=0
FlightPlanCruising Altitude	7.	cruising_altitude=34000
See id detail below	8.	departure_id=VDPP, N11* 32.52', E104* 51.16', +000040.00
	9.	departure_position=PARKING 1
See id detail below	10.	destination_id=WSSS, N1* 19.75', E103* 59.11', +000021.99
FlightPlanDepartureName	11.	departure_name=Pochentong Intl
FlightPlanDestinationName	12.	destination_name=Changi
See id detail below	13.	**alternate_id=WIKB, N1* 07.13', E104* 06.85', +000126.99**
FlightPlanAlternateName	14.	**alternate_name=Hang Nadim**
Flight Play Waypoint Info	15.	waypoint.0=, VDPP, , VDPP, A, N11* 32.52', E104* 51.16', +000040.00,
	16.	waypoint.1=, WSSS, , WSSS, A, N1* 19.75', E103* 59.11', +000021.99,

alternate_id=WIKB, N1* 07.13', E104* 06.85', +000126.99

FlightPlanAlternateAirportIdent	WIKB
FlightPlanAlternateLatitude	N1* 07.13'
FlightPlanAlternateLongitude	E104* 06.85'
FlightPlanAlternateAltitude	+000126.99

The fs9gps values showing the Alternate Airport that has been added:

```
GPS Viewer 1.2                  11111111112
GET SET SLEN 12345678901234567890 STRING NUMBER
 G        12  VDPP to WSSS          FlightPlanTitle, string
 G        10  VDPP, WSSS            FlightPlanDescription, string
 G        4   VDPP                  FlightPlanDepartureAirportIdent, string
 G        15  Pochentong Intl       FlightPlanDepartureName, string
 G        6     11.5419998          FlightPlanDepartureLatitude, degrees
 G        6    104.8526665          FlightPlanDepartureLongitude, degrees
 G        6   40.000004             FlightPlanDepartureAltitude, feet
 G        4   WSSS                  FlightPlanDestinationAirportIdent, string
 G        6   Changi                FlightPlanDestinationName, string
 G        6     1.3291667           FlightPlanDestinationLatitude, degrees
 G        6    103.9851664          FlightPlanDestinationLongitude, degrees
 G        6   21.000009             FlightPlanDestinationAltitude, feet
 G        0     WIKB                FlightPlanAlternateAirportIdent, string
 G        0   Hang Nadim            FlightPlanAlternateName, string
 G        6     1.1188332           FlightPlanAlternateLatitude, degrees
 G        6    104.1141665          FlightPlanAlternateLongitude, degrees
 G        6   126.000001            FlightPlanAlternateAltitude, feet
```

The FlightPlanAlternate variables are not truly *functional*, however. They don't "do" anything. If desired, an Airport ICAO could be easily constructed from the AlternateAirportIdent as follows:

```
'A_ _ _ _ _ _'
(C:fs9gps:FlightPlanAlternateAirportIdent) scat
```

The first line is an A followed by 6 spaces. The xml above yields 'A WIKB', which is the full ICAO for Hang Nadim Airport. I suppose one could then use it to specify a new approach Airport. In a similar manner, FlightPlanAlternateLatitude and Longitude could be used to specify a new Waypoint that could be added to the Flight Plan.

However, when coding a gps or flight management computer from scratch, I guess it would be easier and more realistic to enter the alternate destination (doesn't have to be an Airport – it could be a Transition Waypoint) Ident while Flight Sim is running, storing the Ident as chr >L:Vars for later use as appropriate.

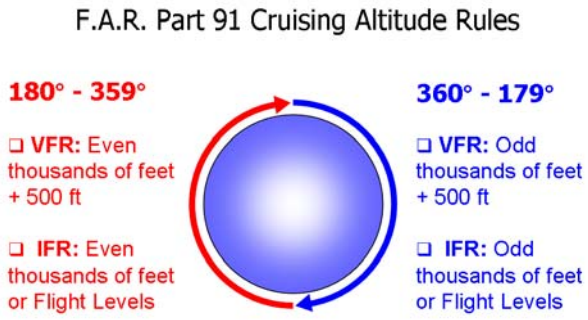
FlightPlanCruisingAltitude DISCUSSION

FlightPlanCruisingAltitude is read only and a separate flight planning application such as the stock FS9 Flight Planner is needed if the user wants a cruising altitude to be computed automatically. Charts and manual entry within FS9 Flight Planner works as well, of course.

Given user inputs of 1) departure airport, 2) destination airport, 3) route type (Direct-GPS, Low Altitude Airways, High Altitude Airways, or VOR to VOR), and 4) flight plan type (VFR or IFR), FS9's Flight Planner determines a flight plan route and then computes a single cruising altitude. The associated gps variables, DepartureAirportIdent, DestinationAirportIdent, RouteType, and FlightPlanType are also read-only.

A detailed discussion of FS9 Flight Planner is beyond the scope of this GPS Guidebook. I'm not certain of the rules that determine exactly how Flight Planner computes Cruising Altitude, however, here are a few Flight Planner notes based on limited observations:

- U.S. F.A.R. Part 91 Cruising Altitude rules are applied world-wide. Above 18,000 feet altitude where, in U.S.A. airspace, flight is governed by Instrument Flight Rules, FS9 Flight Planner still adds 500 feet to the cruising altitude of a flight that is set up as a VFR flight.



- FS9 Flight Planner 'scans' at least a coarse terrain grid along its computed flight route and then provides an amount of clearance above the maximum ground elevation in the calculation of Cruising Altitude.

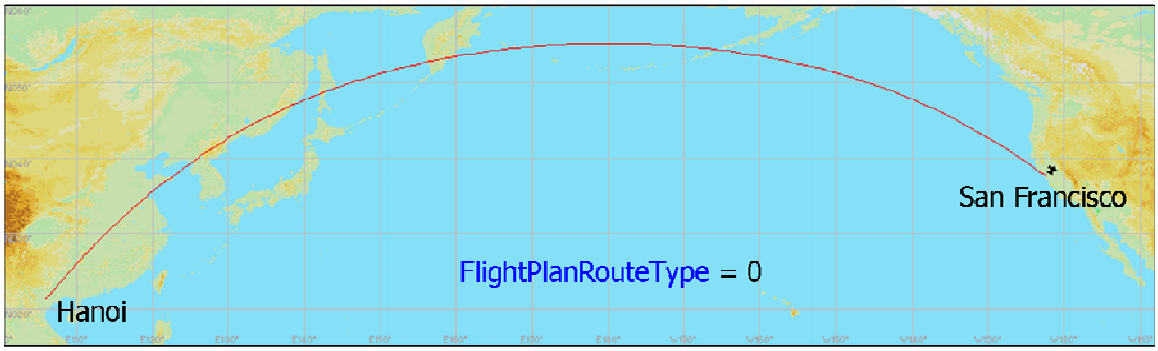
- Ground clearance rules appear to vary geographically. Ground clearance in the USA seems to be 3000 to 4000 feet above highest ground elevation. Flying from France to Italy over the Alps, 7000 feet. Italy to Austria, 4000 feet. Thailand, 3000 feet. And so on. I have not figured out how to predict the value.

- Cruising Altitude often exceeds even the maximum airway segment MEA for flight routes along low altitude Airways.

- FS9 Flight Planner calculates Cruising Altitude once the "Find Route" button is clicked. Subsequent changes to the flight route by adding or deleting waypoints will not automatically re-compute the Cruising Altitude, even if it should.

- Restricted Airspace sometimes influences FS9 Flight Planner's Cruising Altitude. The effect seems to be inconsistent, however, so again, I am not sure.

- As everyone knows, Flight Planner calculates GPS-Direct routes along Great Circle paths.



In summary, when it comes to FS9 Flight Planner's rules for computing CruisingAltitude, I am guessing. At any rate, after you consult your charts, FlightPlanCruisingAltitude is manually settable from within FS9 Flight Planner and can be changed during flight by request to ATC.

FLIGHT PLAN STATUS VARIABLES

❑ FlightPlanIsActiveFlightPlan (bool) [Get]

FlightPlanIsActiveFlightPlan =1 means a flight plan is loaded.

❑ FlightPlanIsLoadedApproach (bool) [Get]

FlightPlanIsLoadedApproach - an approach is loaded (flight plan may/may not be loaded)

❑ FlightPlanIsActiveApproach (bool) [Get, Set]

FlightPlanIsActiveApproach is used to activate a loaded approach. The xml:

```
1 (>C:fs9gps:FlightPlanIsActiveApproach)
```

It requires an argument. 1 for activate, as above. 0 means approach is loaded, but not activated.

❑ FlightPlanIsActiveWaypoint (bool) [Get, ~~Set~~]

FlightPlanIsActiveWaypoint is a bool representing whether or not any Waypoint in the Flight Plan is active. At the departure airport and en route, one waypoint is always active. The status changes upon reaching the destination waypoint, however, at which point FlightPlanIsActiveWaypoint becomes 0.

FlightPlanIsActiveWaypoint is 1 as long as there is a Flight Plan or Approach waypoint that is active.

FlightPlanIsActiveWaypoint appears to be read-only, contrary to the SDK description.

❑ FlightPlanIsDirectTo (bool) [Get, ~~Set~~]

FlightPlanIsDirectTo is a bool representing Direct To status. 1 = Flight Plan is Direct To. 0 = Not Direct To.

FlightPlanIsDirectTo appears to be read-only, contrary to the SDK description.

❑ FlightPlanDirectToWaypoint (enum) [Get]

DirectToWaypoint is the Flight Plan Waypoint Index that the Direct To points to. Because a DirectTo Flight Plan is (apparently) always a two-waypoint Flight Plan, then the only logical DirectToWaypoint value is 1. Indeed, DirectToWaypoint is always 1 when the Flight Plan is DirectTo, or -1 when it is not.

❑ FlightPlanActiveWaypoint (enum) [Get, Set]

FlightPlanActiveWaypoint is the Index number of the waypoint that the aircraft is currently flying directly toward or on an intercept course toward; it's the next waypoint. The active Flight Plan *leg* is the flight path from the previous waypoint to the active Waypoint.

In the stock gps_500 gauge, the Ident of the ActiveWaypoint, if an Ident exists, is displayed in magenta text and the line color of the active Flight Plan leg is also magenta. In the FS9 Map, the line color of the active Flight Plan leg is magenta.

FlightPlanActiveWaypoint is read *and write* capable, contrary to the SDK description.

❑ FlightPlanActiveApproachWaypoint (enum) [Get]

FlightPlanActiveApproachWaypoint is the index number of the active (current) approach segment. ApproachWaypoint does not change when a sub-segment changes – only when the segment changes.

❑ FlightPlanIsActiveWaypointLocked (bool) [Get, Set]

When set to 1, FlightPlanIsActiveWaypointLocked locks the ActiveWaypoint's Index number so that it cannot advance by 1 when the aircraft reaches the Active Waypoint. This has significant consequences – it effectively terminates the Flight Plan at the ActiveWaypoint until and unless ActiveWaypointLocked is set to zero.

If the aircraft is being controlled by a typical autopilot, it will turn 360° upon reaching the locked Waypoint, circling around to repeatedly cross it. The same occurs when an aircraft reaches the destination point of a Flight Plan but does not land (I am referring to *Flight Plan* – not an Approach with a Missed Approach Procedure).

FlightPlanIsActiveWaypointLocked can be set to one or zero at any point in time, locking the ActiveWaypoint or unlocking it and allowing the Flight Plan to continue as normal. When a Flight Plan is opened, FlightPlanIsActiveWaypointLocked is set to zero. Perhaps it is possible outside of xml to create a flight pan with a locked ActiveWaypoint, but I don't know what purpose that would serve.

Note: When adding or deleting Waypoints *beyond* the ActiveWaypoint, fs9gps sets ActiveWaypointLocked to 1, so you may want to subsequently reset it to zero unless you want the Flight Plan to terminate at the ActiveWaypoint.

An example of the use and effects of FlightPlanIsActiveWaypointLocked is given in Example 3 of this section.

❑ FlightPlanWaypointsNumber (enum) [Get]

FlightPlanWaypointsNumber is the total number of waypoints in the flight plan. The Departure airport is always the first waypoint and has Index = 0. The total number of flight plan legs is FlightPlanWaypointsNumber minus 1.

NEWWAYPOINT GROUP: CREATING AND EDITING A FLIGHT PLAN

CREATING A FLIGHT PLAN WITH XML

Flight Plans can be easily created using xml, but unfortunately, not saved owing to a current lack of file I/O capability in the xml environment. Even saving the Flight after editing the flight plan using the NewWaypoint variables will not save the Flight Plan; the gps engine will reject the edited Flight Plan when the Flight is loaded.

To create a new Flight Plan when one is not currently loaded, FlightPlanDirectToDestination is used. Use of the FlightPlanDirectToDestination variable is reviewed later in this section.

Of course, an easy way of editing *and saving* a flight plan by adding or deleting existing navaid or published intersection waypoints is with FS9's built-in Flight Planner. With the Find Route map open, just click on the flight path between any two waypoints and drag it to the new waypoint facility. To delete a waypoint, select the waypoint from the waypoint list to the right of the map, then click "Delete". Then "Save". Very easy.

EDITING A FLIGHT PLAN

The NewWaypoint variables are a small group of Set-only variables that can be used to edit flight plans by adding or deleting en route or alternate destination waypoints, one waypoint at a time.

Adding a new waypoint to an active flight plan is a two step process that involves defining the latitude and longitude of the new waypoint to be added, followed by assigning the waypoint index of the new waypoint (where the new en route waypoint will be inserted in the Flight Plan).

Restating this, the required information for a new en route waypoint is:

1. Latitude and Longitude
2. New Waypoint Index position

A valid en route waypoint can be just a point on the map, not associated with an existing navaid or fs9gps waypoint. Fs9gps will assign WaypointType = 5 (User) to any waypoint added using AddWaypoint that is not an existing navaid or published waypoint.

ENTERING NEW WAYPOINT LATITUDE AND LONGITUDE

There are a few approaches to this:

1. Enter the new waypoint lat and lon directly (necessary for user-defined waypoints)
2. Enter the new waypoint ICAO, from which lat and lon will automatically be accessed by fs9gps

    a. Enter the 12 character ICAO directly (usually not very realistic).

    b. Enter the new waypoint facility (airport, navaid or intersection) Ident followed by an ICAO search that determines the unique ICAO, from which lat and lon will be automatically accessed.

    c. For airport waypoints, enter the airport Name followed by Name Search which can be used to find the airport ICAO.

❑ FlightPlanNewWaypointLatitude
❑ FlightPlanNewWaypointLongitude (degrees) [Set]

Latitude and Longitude of the waypoint to be added. Either NewWaypointLatitude and Longitude, or NewWaypointICAO must be entered before AddWaypoint is executed.

```
51.3278 (>C:fs9gps:FlightPlanNewWaypointLatitude, degrees)
 7.1770 (>C:fs9gps:FlightPlanNewWaypointLongitude, degrees)
```

or

```
'VED    BAM' (>C:fs9gps:FlightPlanNewWaypointICAO)
```

Both will define a new en route waypoint at the same location.

❑ FlightPlanNewWaypointICAO (string) [Set]

FlightPlanNewWaypointICAO can be entered instead of Latitude and Longitude. From the ICAO, fs9gps will automatically access the Ident, Waypoint Type, Latitude and Longitude of the new waypoint. Waypoint Altitude will not be returned because, at least in FS9, you must enter the related facility Group to access altitude.

❑ FlightPlanNewWaypointIdent (string) [Set]

Asking the user to enter the 12 character ICAO isn't a completely realistic sim experience. Instead, entering the Ident of a navaid, intersection, or airport is a more real-world procedure. The ultimate objective is still to define the Lat and Lon of the new waypoint.

Because Idents other than airports are not always unique, using NewWaypointIdent will require an ICAO Search to isolate the correct ICAO from which Lat and Lon will be accessed.

The ICAO Search process is more straight-forward than it may sound and is discussed in the ICAO Search chapter. If the search goal is to display the list of ICAOs containing the specified Ident that the user can select from, then the ICAOSearch code will be quite simple.

Once ICAO Search is complete and the desired ICAO selected, then all that remains is to transfer the ICAOSearchCurrentIcao to FlightPlanNewWaypointIcao:

```
(@c:IcaoSearchCurrentIcao) (>@c:FlightPlanNewWaypointIcao)
```

A more complete xml example is included in the ICAO Search Data Group chapter.

ASSIGNING A WAYPOINT INDEX AND ADDING THE NEW WAYPOINT

❑ FlightPlanAddWaypoint (enum) [Set]

FlightPlanAddWaypoint is used to add a single, additional waypoint to a loaded flight plan, one added waypoint at a time. It requires an argument (index pointer) to indicate where in the Flight Plan the new waypoint will be inserted.

```
2 (>C:fs9gps:FlightPlanAddWaypoint)
```

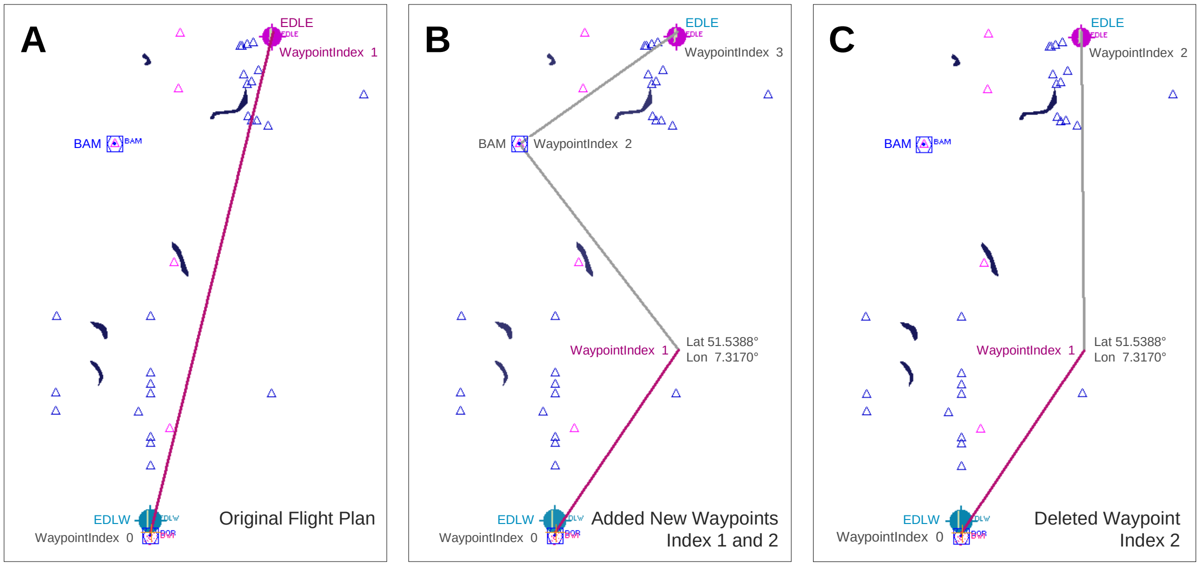
will add a new waypoint at WaypointIndex 2 with whatever Lat and Lon had previously been specified using NewWaypointLatitude and Longitude, or NewWaypointICAO. The waypoint previously at WaypointIndex 2 is advanced to become WaypointIndex 3. 3 becomes 4, and so forth. New waypoints can only be added/deleted to Flight Plans, not to Approaches.

❑ FlightPlanDeleteWaypoint (enum) [Set]

FlightPlanDeleteWaypoint is used to delete a single waypoint from a loaded flight plan. One waypoint delete at a time. It requires an argument (index pointer) to indicate which waypoint is to be deleted.

Example 1: FlightPlanAddWaypoint and FlightPlanDeleteWaypoint

The following example demonstrates FlightPlanAddWaypoint and FlightPlanDeleteWaypoint:



Map **A** shows a Direct To routing from Dortmund Airport, Dortmund Germany to Essen-Mülheim Airport, Essen/Mülheim Germany. The table below lists the FlightPlanWaypoint variables:

EDIT FLIGHT PLAN

FlightPlanTitle: EDLW to EDLE
2 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
0 :FlightPlanRouteType 2 :FlightPlanFlightPlanType

------------------------------ FlightPlanWaypoint ------------------------------

Idx	ICAO	111 123456789012	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	EDLW	EDLW	424	1	0	51.52330	7.62637	0.00	0.0	27.1	0.0	0.00	0.00	0.00	12.61	240.3	0.0	0.0
1	A	EDLE	EDLE	424	1	256	51.40018	6.92987	27.07	27.1	0.0	26.7	8.12	0.00	14.63	12.86	0.0	5.9	0.0

Next, two new waypoints are added using NewWaypoint variables. The xml:

```
<!-- The first new Waypoint -->
   'VED    BAM' (>C:fs9gps:FlightPlanNewWaypointICAO)
   1 (>C:fs9gps:FlightPlanAddWaypoint)

<!-- The second new Waypoint -->
   51.5388 (>C:fs9gps:FlightPlanNewWaypointLatitude, degrees)
   7.3170 (>C:fs9gps:FlightPlanNewWaypointLongitude, degrees)
   1 (>C:fs9gps:FlightPlanAddWaypoint)
```

The new routing is shown in Map **B**, and the new FlightPlanWaypoint variable list is shown below. The red outline highlights the new waypoints. Note that the User-defined waypoint, WaypointIndex 1 (just a lat, lon position and not an existing fs9gps facility), does not have an ICAO. Note also that both new waypoints are added using the same argument (i.e., 1) for FlightPlanAddWaypoint. When the second waypoint is added also as Waypoint 1, the previously existing Waypoint 1 is automatically advanced to become Waypoint 2. 2 becomes 3, and so forth.

```
EDIT FLIGHT PLAN

FlightPlanTitle: EDLW to EDLE
   4 :FlightPlanWaypointsNumber   1 :FlightPlanActiveWaypoint
   0 :FlightPlanRouteType         2 :FlightPlanFlightPlanType
```

FlightPlanWaypoint

Idx	ICAO 123456789012	111	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	EDLW	EDLW	424	1	0	51.52330	7.62637	0.00	0.0	35.5	0.0	0.00	0.00	0.00	0.00	240.2	0.0	0.0
1				0	5	276	51.53880	7.31700	11.59	11.6	23.9	11.6	3.47	0.00	0.00	0.00	0.0	2.5	0.0
2	VED	BAM	BAM	0	3	204	51.32776	7.17699	13.70	25.3	10.2	13.7	4.10	0.00	0.00	0.00	0.0	3.0	0.0
3	A	EDLE	EDLE	424	1	297	51.40018	6.92987	10.23	35.5	0.0	10.2	3.07	0.00	0.00	0.00	0.0	2.2	0.0

Upon executing AddWaypoint, fs9gps automatically updates waypoint index, magnetic heading, distances, ETEs, ETAs and fuel variables for all affected waypoints.

Finally, WaypointIndex 2 (BAM VOR-DME) is deleted using FlightPlanDeleteWaypoint. The xml:

```
2 (>C:fs9gps:FlightPlanDeleteWaypoint)
```

The new routing is shown in Map **C**, and the final FlightPlanWaypoint variable list is shown below:

```
EDIT FLIGHT PLAN

FlightPlanTitle: EDLW to EDLE
   3 :FlightPlanWaypointsNumber   1 :FlightPlanActiveWaypoint
   0 :FlightPlanRouteType         2 :FlightPlanFlightPlanType
```

FlightPlanWaypoint

Idx	ICAO 123456789012	111	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	EDLW	EDLW	424	1	0	51.52333	7.62633	0.00	0.0	28.3	0.0	0.00	0.00	0.00	0.00	241.9	0.0	0.0
1				0	5	276	51.53880	7.31700	11.58	11.6	16.7	11.6	3.47	0.00	0.00	0.00	0.0	2.5	0.0
2	A	EDLE	EDLE	424	1	242	51.40017	6.92983	16.69	28.3	0.0	16.7	5.00	0.00	0.00	0.00	0.0	3.6	0.0

❑ FlightPlanDirectToDestination (bool) [Set]

FlightPlanDirectToDestination will create a new, two-waypoint Flight Plan originating at the aircraft's current x,y,z position and culminating at the latitude and longitude defined by FlightPlanNewWaypointLatitude and Longitude, or FlightPlanNewWaypointICAO.

If no Flight Plan is currently loaded, FlightPlanDirectToDestination will create a new two-waypoint Flight Plan. If a Flight Plan is currently active, FlightPlanDirectToDestination will replace the entire Flight Plan with the new two-waypoint one.

The current aircraft location will become WaypointIndex 0. FlightPlanWaypointAltitude will be set to the current aircraft altitude and FlightPlanWaypointType will be 5 (User). No ICAO will be associated with this waypoint.

The Direct To location can be any latitude and longitude; it does not have to be an fs9gps navaid facility or intersection, nor a waypoint currently in the Flight Plan. However, FlightPlanNewWaypointLatitude and Longitude, or FlightPlanNewWaypointICAO must be defined immediately preceding the FlightPlanDirectToDestination statement as follows.

Direct To a custom, user-defined lat and lon:

```
37.8487 (>C:fs9gps:FlightPlanNewWaypointLatitude, degrees)
-97.8157 (>C:fs9gps:FlightPlanNewWaypointLongitude, degrees)
(>C:fs9gps:FlightPlanDirectToDestination)
```

L:Vars could be substituted for the numbers, of course:

```
(L:DTO_Lat, degrees) (>@c:FlightPlanNewWaypointLatitude, degrees)
```

FlightPlanDirectToDestination does not require an argument.

Direct To a Waypoint in the Flight Plan:

```
(L:DTOWaypointIndex, enum) (>@c:FlightPlanWaypointIndex)
(@c:FlightPlanWaypointICAO) (>@c:FlightPlanNewWaypointICAO)
(>@c:FlightPlanDirectToDestination)
```

In the example above, `L:DTOWaypointIndex` is a user-specified Flight Plan waypoint index number. It is entered into WaypointIndex from which WaypointICAO is determined. From there, the ICAO transfer into NewWaypointICAO will provide the latitude and longitude the DirectToDestination statement needs.

Note that this approach requires that the waypoint have an ICAO. All fs9gps facilities (airports, navaids, waypoints/intersections) have a unique ICAO, but if the flight plan contains a custom, user-defined waypoint, that waypoint will not have an ICAO.

Direct To any fs9gps Facility:

This approach requires that the facility ICAO be determined as the first step. Any source of the ICAO will do:

- ICAOSearchCurrentICAO
- NameSearchCurrentICAO
- Waypoint or Facility Group ICAOs such as WaypointAirportICAO
- Nearest Group ICAOs such as NearestVorCurrentICAO

This is followed by the ICAO transfer into NewWaypointICAO and, finally, the DirectToDestination statement:

```
(@c:NearestVorCurrentICAO) (>@c:FlightPlanNewWaypointICAO)
(>@c:FlightPlanDirectToDestination)
```

Note: Following execution of FlightPlanDirectToDestination, it appears that use of gps variables that attempt to access the ICAO or Ident of either of the DirectTo Waypoints, or properties of those Waypoints, may cause the simulation to crash.

As an example, the following operations will cause the sim to crash if they are preceeded by execution of FlightPlanDirectToDestination:

- `((C:fs9gps:FlightPlanDescription) slen)`
- `((C:fs9gps:FlightPlanDepartureAirportIdent) slen)`
- `((C:fs9gps:FlightPlanDepartureName) slen)`
- `((C:fs9gps:FlightPlanDestinationAirportIdent) slen)`
- `((C:fs9gps:FlightPlanDestinationName) slen)`

However, use of those variables without slen will just return an empty string without causing a crash.

Admittedly, I do not yet understand the issue very well and there obviously is more going on than I realize. I suspect that other operations will also cause a sim crash. ***Suffice it to say that I have found that a simulation crash can easily occur following FlightPlanDirectToDestination***, so be on the alert.

❑ FlightPlanCancelDirectTo (bool) [Set]

FlightPlanCancelDirectTo restores the Flight Plan to the state prior to execution of FlightPlanDirectToDestination. However, if the Flight Plan is changed (AddWaypoint or DeleteWaypoint) after DirectToDestination is executed, then FlightPlanCancelDirectTo will no longer be able to restore the Flight Plan to the prior state.

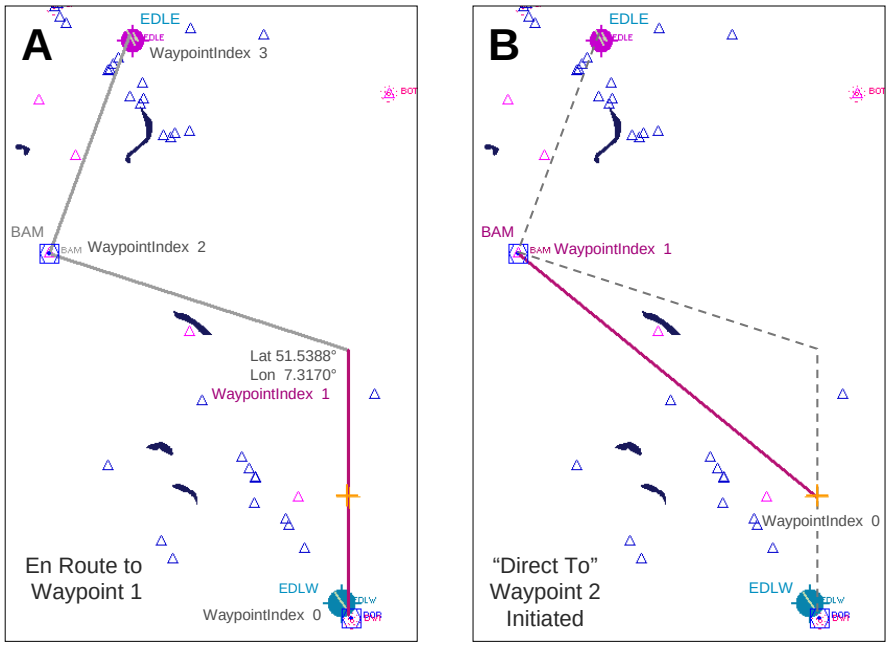
The xml:

```
(>C:fs9gps:FlightPlanCancelDirectTo)
```

FlightPlanCancelDirectTo does not require an argument.

Example 2: FlightPlanDirectToDestination and CancelDirectTo

The following example demonstrates FlightPlanDirectToDestination and CancelDirectTo:



Map **A** shows flight progress as the aircraft is en route to waypoint 1. The aircraft position is indicated with the orange colored + symbol. The table below lists the FlightPlanWaypoint variables.

```
EDIT FLIGHT PLAN

FlightPlanTitle: EDLW to EDLE
   4 :FlightPlanWaypointsNumber   1 :FlightPlanActiveWaypoint
   0 :FlightPlanRouteType         2 :FlightPlanFlightPlanType
```

FlightPlanWaypoint

Idx	ICAO 111	ICAO 123456789012	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	EDLW	EDLW	424	1	0	51.52333	7.62633	0.00	0.0	35.5	0.0	0.00	0.00	0.00	12.80	240.9	0.0	0.0
1				0	5	276	51.53880	7.31700	11.58	11.6	23.9	6.4	3.47	0.00	2.47	12.87	0.0	2.5	0.0
2	VFD	BAM	BAM	0	3	204	51.32776	7.17699	13.70	25.3	10.2	13.7	4.10	0.00	5.30	12.96	0.0	3.0	0.0
3	A	EDLE	EDLE	424	1	297	51.40017	6.92983	10.23	35.5	0.0	10.2	3.07	0.00	3.95	13.03	0.0	2.2	0.0

Map **B** shows the Flight Plan immediately after a DirectToDestination Waypoint 2 is initiated.

The xml:

```
(L:DTOWaypointIndex, enum) (>@c:FlightPlanWaypointIndex)
(@c:FlightPlanWaypointICAO) (>@c:FlightPlanNewWaypointICAO)
(>@c:FlightPlanDirectToDestination)
```

where the user has entered 2 for L:DTOWaypointIndex. The table below lists the FlightPlanWaypoint variables. Note that the present aircraft position becomes the new Waypoint 0, the aircraft's current altitude (2964') becomes WaypointAltitude for Index 0,

and the WaypointType is 5 (User). The DirectTo Waypoint (original Waypoint 2, BAM VOR-DME) is now WaypointIndex 1, and WaypointMagneticHeading and Distance variables are adjusted. FlightPlanTitle is also updated to reflect the DTO.

```
EDIT FLIGHT PLAN

FlightPlanTitle: Direct to BAM
   2 :FlightPlanWaypointsNumber   1 :FlightPlanActiveWaypoint
   0 :FlightPlanRouteType         0 :FlightPlanFlightPlanType
```

FlightPlanWaypoint

Idx	ICAO 111 123456789012	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0			2964	5	0	51.53029	7.48828	0.00	0.0	16.8	0.0	0.00	0.00	0.00	12.83	239.7	0.0	0.0
1	VED	BAM	0	3	226	51.32776	7.17699	16.83	16.8	0.0	16.8	5.03	0.00	6.50	12.94	0.0	3.7	0.0

Map **C** shows progress of the flight en route to the Direct To waypoint. At this point, a CancelDirectTo is intitated. The xml:

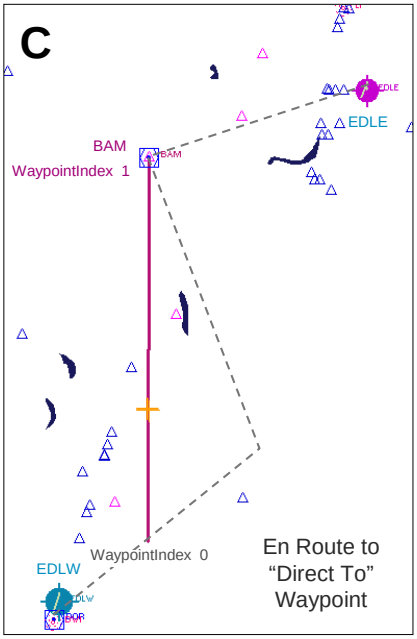
```
(>@c:FlightPlanCancelDirectTo)
```

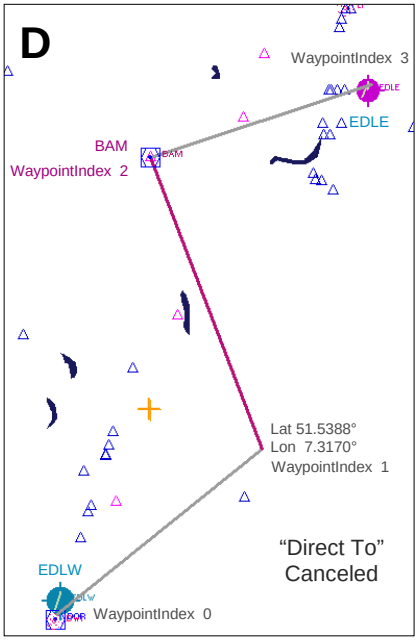
Map **D** reflects the Flight Plan immediately after CancelDirectTo, and the table below lists the FlightPlanWaypoint variables. Note that the original Flight Plan is restored and that distance and fuel variables associated with ActiveWaypoint 2 are updated. The ActiveWaypoint is now 2, and, if on autopilot, the aircraft will begin a right turn to intercept the Waypoint 1 to Waypoint 2 leg.

```
EDIT FLIGHT PLAN

FlightPlanTitle: EDLW to EDLE
   4 :FlightPlanWaypointsNumber   2 :FlightPlanActiveWaypoint
   0 :FlightPlanRouteType         2 :FlightPlanFlightPlanType
```

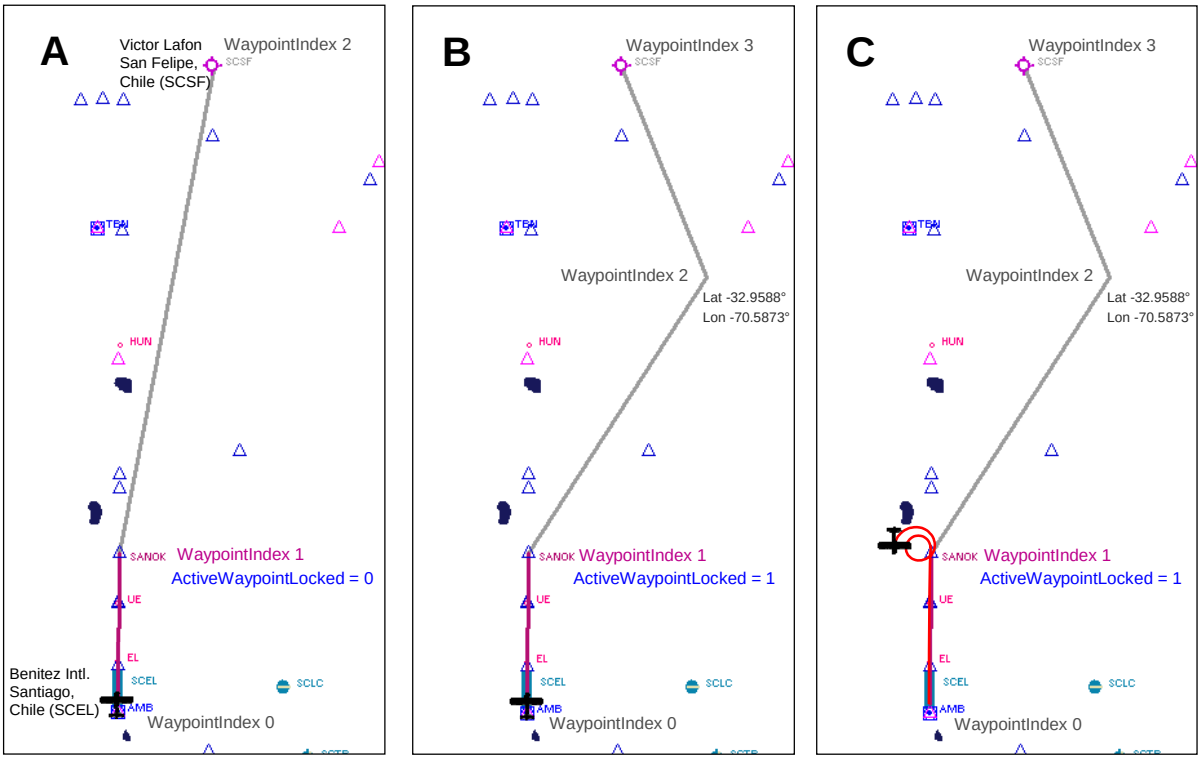
FlightPlanWaypoint

Idx	ICAO 111 123456789012	Ident	Alt	Type	Mag Hdg	Lat	Lon	Dist	Dist Ttl	Dist Rem	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	EDLW	EDLW	424	1	0	51.52333	7.62633	0.00	0.0	35.5	0.0	0.00	0.00	12.87	238.0	0.0	0.0
1				0	5	276	51.53880	7.31700	11.58	11.6	23.9	0.0	3.47	0.00	0.00	12.87	238.0	2.5	0.0
2	VED	BAM	BAM	0	3	204	51.32776	7.17699	13.70	25.3	10.2	5.3	4.10	0.00	3.40	12.92	0.0	3.0	0.0
3	A	EDLE	EDLE	424	1	297	51.40017	6.92983	10.23	35.5	0.0	10.2	3.07	0.00	3.15	12.98	0.0	2.2	0.0





Example 3: ActiveWaypointLocked, AddWaypoint and ActiveWaypoint



This example begins with a Flight Plan from Arturo Merino Benitez Intl. Airport (SCEL), Santiago, Chile, to Victor Lafon Airport (SCSF), San Felipe, Chile. It includes WaypointIndex 1, SANOK Intersection. The Flight Plan map is shown in Figure **A**, above, and the table below lists some of the Flight Plan variables. Note that ActiveWaypoint is 1 and ActiveWaypointLocked is 0. The Flight Plan was created in FS9's Flight Planner.

```
FLIGHT PLAN

FlightPlanTitle: SCEL to SCSF
       3 :FlightPlanWaypointsNumber          1 :FlightPlanActiveWaypoint     0 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType                2 :FlightPlanFlightPlanType     1 :FlightPlanIsActiveWaypoint

------------------------------------------- FlightPlanWaypoint -------------------------------------------
     ICAO      111                          Mag                                    Dist      Dist      Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat         Lon       Dist       Ttl       Rem      Dist      ETE
0    A       SCEL    SCEL  1554    1    0  -33.40935   -70.78492     0.00       0.0      39.9       0.0     0.00
1    WSCSCELSANOK  SANOK     0    2  352  -33.25284   -70.79138     9.40       9.4      30.5       9.4     2.82
2    A       SCSF    SCSF  2160    1    3  -32.74934   -70.70198    30.55      39.9       0.0      30.5     9.15
```

Figure **B** shows the results of adding a user-defined waypoint as new WaypointIndex 2 at Latitude S32° 57.53', Longitude W70° 35.238'. The xml:

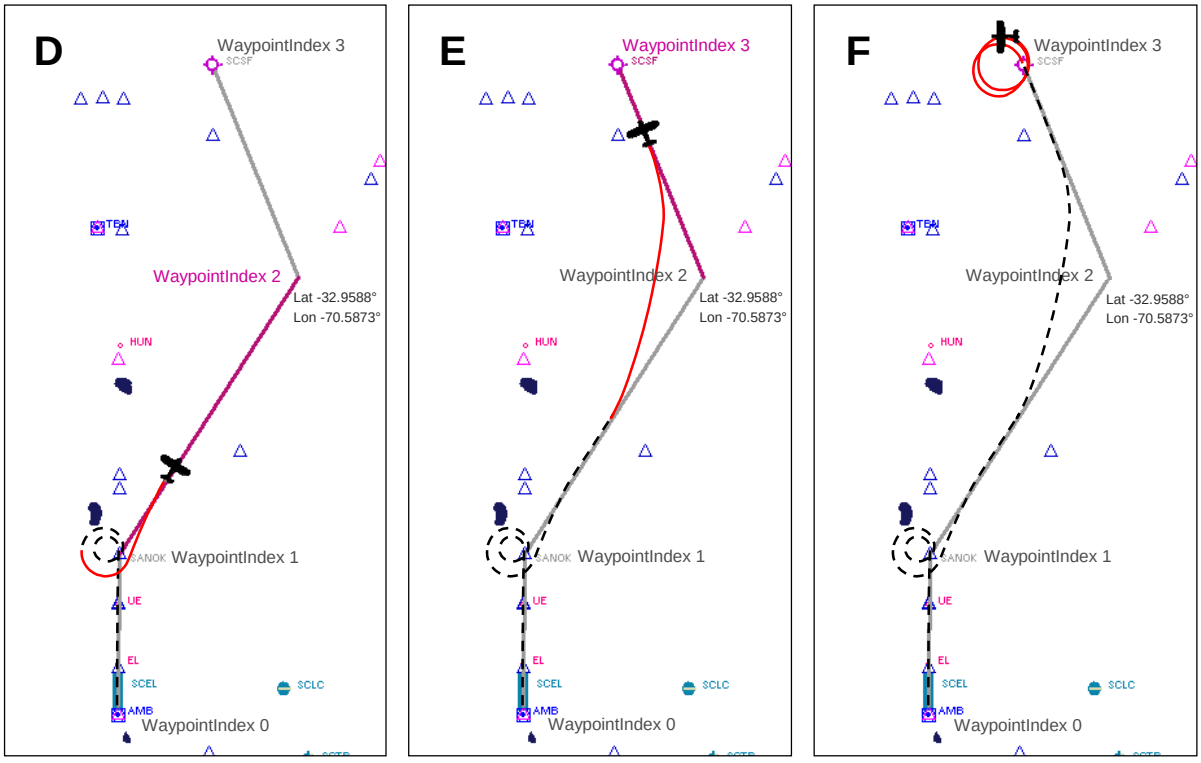
```
-32.95883 (>C:fs9gps:FlightPlanNewWaypointLatitude, degrees)
-70.58733 (>C:fs9gps:FlightPlanNewWaypointLongitude, degrees)
2 (>C:fs9gps:FlightPlanAddWaypoint)
```

Note from the table below that ActiveWaypointLocked is now 1 as a result of the AddWaypoint statement. The destination waypoint, SCSF, which had been WaypointIndex 2

has been automatically advanced to become WaypointIndex 3 and the associated distances and time have been adjusted to accommodate the new waypoint.

```
FLIGHT PLAN

FlightPlanTitle: SCEL to SCSF
       4 :FlightPlanWaypointsNumber          1 :FlightPlanActiveWaypoint     1 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType                2 :FlightPlanFlightPlanType     1 :FlightPlanIsActiveWaypoint

------------------------------------------- FlightPlanWaypoint -------------------------------------------
     ICAO     111                          Mag                                  Dist      Dist      Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat         Lon      Dist        Ttl       Rem       Dist      ETE
0    A      SCEL    SCEL  1554    1    0  -33.40935  -70.78492      0.00        0.0      43.6       0.0     0.00
1    WSCSCELSANOK  SANOK     0    2  352  -33.25284  -70.79138      9.40        9.4      34.2       9.4     2.82
2                             0    5   24  -32.95880  -70.58730     20.41       29.8      13.8      20.4     6.12
3    A      SCSF    SCSF  2160    1  330  -32.74934  -70.70198     13.83       43.6       0.0      13.8     4.13
```

Figure **C** shows the flight at Waypoint 1. Because ActiveWaypointLocked is 1, the Active Waypoint is locked and does not advance to the next Index number. The result is that the aircraft (controlled by an autopilot) keeps turning 360° to repeatedly intercept Waypoint 1.



Mid-way through the second 360° turn, ActiveWaypointLocked is reset to zero:

```
0 (>C:fs9gps:FlightPlanIsActiveWaypointLocked)
```

and as shown in Figure **D**, ActiveWaypoint advances to WaypointIndex 2, and the aircraft turns to intercept the flight leg to Waypoint 2.

At about the half way point to Waypoint 2 (Figure **E**), the ActiveWaypoint is changed to 3 by user input:

```
3 (>C:fs9gps:FlightPlanActiveWaypoint)
```

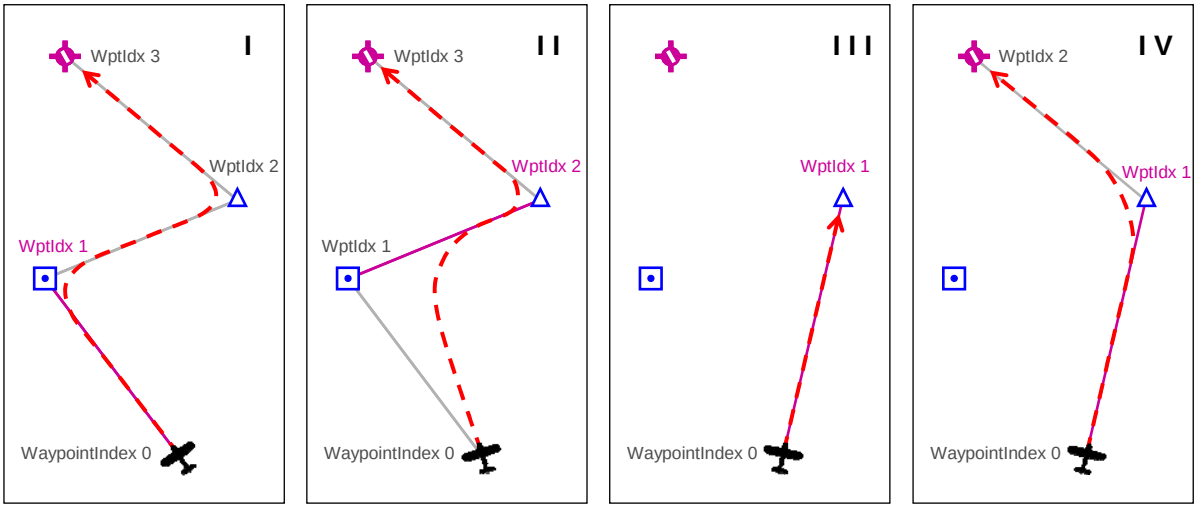
The aircraft now turns to intercept the flight leg from Waypoint 2 to Waypoint 3, bypassing Waypoint 2. The Flight Plan remains unchanged as shown in the table below. ActiveWaypointLocked remains 0.

```
FLIGHT PLAN

FlightPlanTitle: SCEL to SCSF
       4 :FlightPlanWaypointsNumber        3 :FlightPlanActiveWaypoint     0 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType              2 :FlightPlanFlightPlanType     1 :FlightPlanIsActiveWaypoint

---------------------------------------------- FlightPlanWaypoint ----------------------------------------------
     ICAO     111                        Mag                                    Dist      Dist      Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat        Lon      Dist       Ttl       Rem      Dist       ETE
0    A      SCEL    SCEL  1554    1    0  -33.40935  -70.78492     0.00       0.0      43.6       0.0      0.00
1    WSCSCELSANOK  SANOK     0    2  352  -33.25284  -70.79138     9.40       9.4      34.2       0.0      2.82
2                          0    5   24  -32.95880  -70.58730    20.41      29.8      13.8       0.0      6.12
3    A      SCSF    SCSF  2160    1  330  -32.74934  -70.70198    13.83      43.6       0.0      21.8      4.13
```

Finally, the aircraft reaches the destination waypoint but does not land (Figure **F**). At this point, under autopilot control, it begins repetitive 360° turns to intercept Waypoint 3 because there is no further waypoint to fly to. As shown in the table below, ActiveWaypoint remains 3, ActiveWaypointLocked remains 0, but now, FlightPlanIsActiveWaypoint has changed from 1 to 0 indicating that there is no longer an active waypoint.

```
FLIGHT PLAN

FlightPlanTitle: SCEL to SCSF
       4 :FlightPlanWaypointsNumber        3 :FlightPlanActiveWaypoint     0 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType              2 :FlightPlanFlightPlanType     0 :FlightPlanIsActiveWaypoint

---------------------------------------------- FlightPlanWaypoint ----------------------------------------------
     ICAO     111                        Mag                                    Dist      Dist      Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat        Lon      Dist       Ttl       Rem      Dist       ETE
0    A      SCEL    SCEL  1554    1    0  -33.40935  -70.78492     0.00       0.0      43.6       0.0      0.00
1    WSCSCELSANOK  SANOK     0    2  352  -33.25284  -70.79138     9.40       9.4      34.2       0.0      2.82
2                          0    5   24  -32.95880  -70.58730    20.41      29.8      13.8       0.0      6.12
3    A      SCSF    SCSF  2160    1  330  -32.74934  -70.70198    13.83      43.6       0.0       0.0      4.13
```

Example 4: Changing the Active Waypoint



ActiveWaypoint can be changed different ways. The examples above demonstrate a Flight Plan from Waypoint 0 to 1 to 2 to 3 (Figure **I**). The aircraft position is Waypoint 0 and the flight path is shown in a red dashed line. The ActiveWaypoint is identified by a magenta flight leg and text color.

In Figure **II**, the Flight Plan has been edited to make Waypoint 2 the ActiveWaypoint:

```
2 (>@c:FlightPlanActiveWaypoint)
```

On autopilot, this will cause the aircraft to intercept the Waypoint 2 leg, by-passing Waypoint 1. Upon reaching Waypoint 2, the aircraft will proceed to the destination point, Waypoint 3, as usual. The Flight Plan is un-altered.

In Figure **III**, the Flight Plan is changed to make Waypoint 2 the DirectTo destination:

```
2 (>@c:FlightPlanWaypointIndex)
(@c:FlightPlanWaypointICAO) (>@c:FlightPlanNewWaypointICAO)
(>@c:FlightPlanDirectToDestination)
```

The aircraft will proceed directly to original Waypoint 2 which is now Waypoint 1. DirectToDestination always creates a two waypoint Flight Plan with the aircraft position as Waypoint 0 and the destination point (termination of the Flight Plan) as Waypoint 1.

Another alternative would be to delete Waypoint 1, as in Figure **IV**:

```
1 (>@c:FlightPlanWaypointIndex)
(@c:FlightPlanWaypointICAO) (>@c:FlightPlanNewWaypointICAO)
(>@c: FlightPlanDeleteWaypoint)
```

When Waypoint 1 is deleted, then previous Waypoint 2 becomes the new Waypoint 1, and so forth.

NEWAPPROACH GROUP: ADDING OR CHANGING AN APPROACH

The NewApproach group is a small group of Set-only variables that can be used to add or change Instrument Approaches and Transitions.

The following information is needed:

- The destination airport. Specifically, the ICAO of the destination airport must be known.
- The desired approach and transition index. Every airport having an Instrument Approach Procedure has an indexed approach list containing at least one approach. The index pointer of the desired approach is the required information. Similarly, the index pointer of the desired approach transition is also required. If omitted, the default index value of zero will be assumed for each.

❑ FlightPlanNewApproachAirport (string) [Set]

FlightPlanNewApproachAirport is the full ICAO of the destination airport for the approach you wish to add or change. The airport can be the same one currently in the Flight Plan or a different one. Any source of the ICAO will do:

- ICAOSearchCurrentICAO
- NameSearchCurrentICAO
- Waypoint or Facility Group ICAOs such as WaypointAirportICAO
- Nearest Group ICAOs such as NearestVorCurrentICAO

An example xml statement:

```
(@c:IcaoSearchCurrentICAO) (>@c:FlightPlanNewApproachAirport)
```

❑ FlightPlanNewApproachApproach (enum) [Set]

FlightPlanNewApproachApproach is the index pointer to the approach you want to add or change. The list of instrument approaches is found in the WaypointAirport Group and can be viewed by looping through WaypointAirportCurrentApproach.

❑ FlightPlanNewApproachTransition (enum) [Set]

FlightPlanNewApproachTransition is the index pointer to the desired transition associated with the desired approach. The list of instrument approach transitions is found in the WaypointAirport Group and can be viewed by looping through WaypointAirportApproachCurrentTransition.

❑ FlightPlanNewApproachMissed (bool) [Set]

FlightPlanNewApproachMissed is used to include or exclude the Missed Approach Procedure in the Approach to be added / edited.

- ❑ 0 The Missed Approach Procedure will be **excluded** from the new Approach Procedure. In this situation, if the aircraft does not land at the NewApproachAirport, then (if most common autopilots are controlling the aircraft) it will proceed to the destination waypoint indicated in the Flight Plan. After crossing the destination waypoint, the aircraft will fly a 360° turn to re-intercept the waypoint and will continue to repeat the 360° turn.
- ❑ 1 The Missed Approach Procedure will be **included** in the new Approach Procedure. FlightPlanNewApproachMissed is a Boolean, so any number other than zero will include the Missed Approach Procedure.

  If omitted, the default value of 1, include Missed Approach Procedure, will be assumed by fs9gps.

❑ FlightPlanNewApproachAddInitialLeg (enum) [Set]

FlightPlanNewApproachAddInitialLeg is used to add an initial approach segment. It facilitates routing the aircraft to the Approach Transition Waypoint. The additional segment starts at either the current aircraft location or the Termination Point of the Flight Plan, and extends to the Transition Waypoint of the Approach. The arguments are:

- ❑ 0 **No initial approach segment** will be added.
- ❑ 1 An initial segment from the **aircraft location to the Transition Waypoint** will be added when the approach is **loaded**.
- ❑ 2 An initial segment from the **Termination Point of the Flight Plan to the Transition Waypoint** will be added when the approach is **loaded**.
- ❑ 3 An initial segment from the **aircraft location to the Transition Waypoint** will be added when the approach is **activated**.
- ❑ 4 An initial segment from the **Termination Point of the Flight Plan to the Transition Waypoint** will be added when the Approach is **activated**.
- ❑ 5+ Same as 0

For all cases, the FlightPlanWaypointApproach list will not contain the new leg as a separate entry, but distances of the first approach leg are adjusted to account for the added initial leg (FlightPlanApproachSegmentLength and Distance, ApproachRemainingTotalDistance).
Note that the Microsoft ESP web page (http://msdn.microsoft.com/en-us/library/cc526954.aspx#FlightPlanData) lists AddInitialLeg units as Unavailable, which might be interpreted to suggest that this variable is inactive. It works fine, however, at least in FS9.

❑ FlightPlanLoadApproach (enum) [Set]

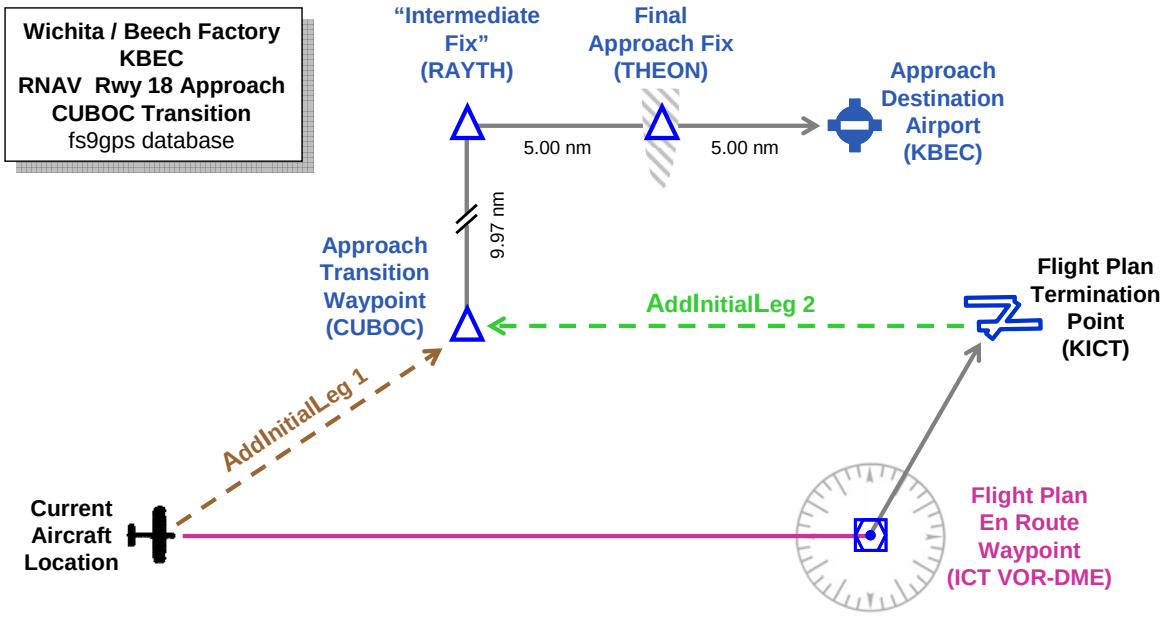
FlightPlanLoadApproach Loads, or Loads and Activates the desired new approach and transition, or Loads and Activates a Vectors-To-Final transition for the current approach depending on the argument used:

- ❑ 0 or negative = **No action**
- ❑ 1 **Load.** The NewApproachAirport, NewApproachApproach, NewApproachTransition, NewApproachMissed and NewApproachAddInitialLeg variables are Loaded but not Activated.
- ❑ 2 **Load and Activate.** The NewApproachAirport, NewApproachApproach, NewApproachTransition, NewApproachMissed and NewApproachAddInitialLeg variables are Loaded and Activated.
- ❑ 3 **Vectors-To-Final.** LoadApproach 3 operates on the existing loaded or activated approach. It will load *and activate* a Vectors-To-Final transition for the currently loaded / activated approach, replacing the existing transition. The new FlightPlanWaypointApproachIndex 0 becomes an extended Final Approach Fix. fs9gps adds 5.00 NMiles to the Final Approach Fix with the same bearing as the Final Approach segment to accommodate the distance that may be required to turn to the Final Approach heading after intercepting the "Vectors-To-Final Fix". You must provide an initial leg or the aircraft will fly to the FAF, not the extended FAF. See the example at the end of this section for further explanation.

  If the intention is to load or activate a *new* approach with a Vectors transition, then the proper choice is to select the new NewApproachAirport and NewApproachApproach, and then NewApproachTransition = 0 (0 is always the Vectors transition Index), followed by NewApproachMissed = 0 or 1, NewApproachAddInitialLeg = 1 or 2, and finally, LoadApproach = 1, 2, or 3.
- ❑ 4+ **Same as 2**

Example 5: Adding or Changing an Approach

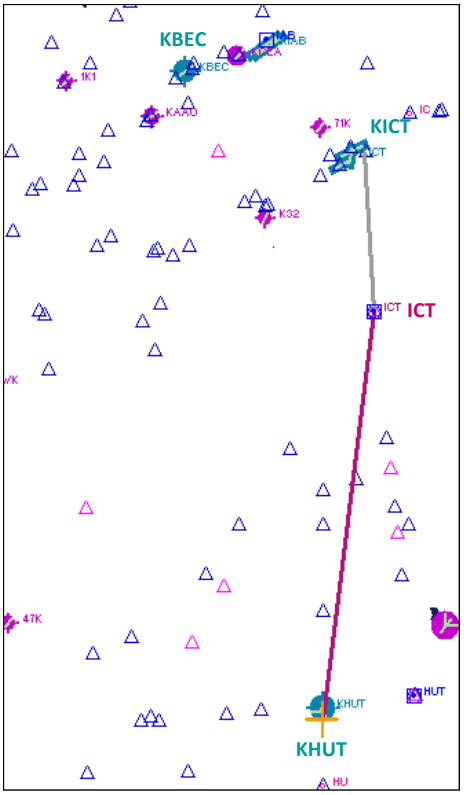
Example 5 demonstrates use of the NewApproach Group variables. It uses the fs9gps RNAV Rwy 18 Approach into the Beech Aircraft Factory Airport (KBEC), Wichita, Kansas, USA. The stock fs9gps database seems not to contain current RNAV Waypoints; instead, the approach is defined using fs9gps terminal waypoints (△). The waypoint names and positions and approach nomenclature used in Example 5 are shown below.



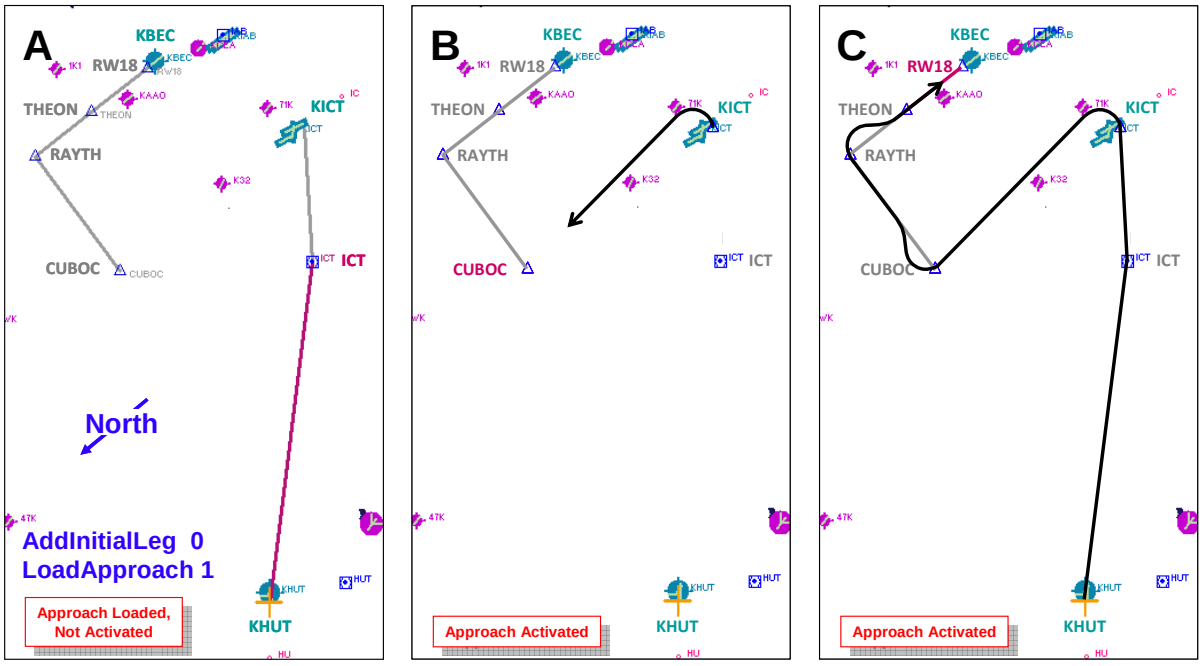
The example begins with a Flight Plan from KHUT to KICT via the ICT VOR-DME en route waypoint.

It is subsequently edited to add the KBEC RNAV18 Approach. In the examples that follow, the Missed Approach Procedure is excluded from the Approach, and, in Examples 5.1 through 5.4, the Approach is loaded, but not activated.

The aircraft does not land at KICT. Instead, it executes the approach into KBEC after passing over KICT, the termination point of the Flight Plan. fs9gps automatically activates a loaded, but not activated, Approach when the aircraft reaches the termination point of the Flight Plan but does not land there.



Example 5.1 NewApproachAddInitialLeg = 0, LoadApproach = 1



In Figure **A**, above, the new approach, KBEC RNAV18, has been added with NewApproachAddInitialLeg set to 0. The xml (the order is important):

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
0 (>@c:FlightPlanNewApproachAddInitialLeg)
1 (>@c:FlightPlanLoadApproach)
```

```
FLIGHT PLAN

FlightPlanTitle: KHUT to KICT
       3 :FlightPlanWaypointsNumber        1 :FlightPlanActiveWaypoint     0 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType              2 :FlightPlanFlightPlanType     1 :FlightPlanIsActiveWaypoint

------------------------------------------- FlightPlanWaypoint -------------------------------------------
     ICAO     111                      Mag                                  Dist      Dist      Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat         Lon      Dist      Ttl       Rem      Dist      ETE
0    A      KHUT    KHUT  1541    1    0   38.07383   -97.87067      0.00      0.0      33.2       0.0     0.00
1    VK3    ICT      ICT  1470    3  138   37.74517   -97.58383     23.95     23.9       9.3      23.9     6.82
2    A      KICT    KICT  1332    1  128   37.63550   -97.44583      9.29     33.2       0.0       9.3     2.60

FLIGHT PLAN NEW APPROACH: KBEC

       4 :ApprWaypointsNumber              6 :FlightPlanApprType           3 :FlightPlanWaypointApproachIndex
 RNAV 18 :FlightPlanApprName             CUBOC :FlightPlanApprTransName    0 :FlightPlanActiveApproachWaypoint
       0 :FlightPlanIsActiveApproach         2 :FlightPlanApproachIndex    5 :FlightPlanApproachTransitionIndex

------------------------------------- FlightPlanWaypointApproach -------------------------------------
     ICAO     111                      Latitude     Longitude    Latitude    Longitude                Leg  Course
Idx  123456789012   Name Type Mode    (Deg Min)     (Deg Min)   (Degrees)   (Degrees)   Alt  Trgt    Dist   (mag)
0    WK3KBECCUBOC  CUBOC    1    1  37 54.1838   -97 22.8128   37.903064  -97.380214  4000     0   0.000   -1.00
1    WK3KBECRAYTH  RAYTH    1    1  37 52.1593   -97 10.4420   37.869322  -97.174033  3000     0   9.972  101.78
2    WK3KBECTHEON  THEON    1    1  37 47.2380   -97 11.5962   37.787300  -97.193269  3000     0   5.006  190.49
3    RK3KBECRW18    RW18    1    2  37 42.3165   -97 12.7477   37.705275  -97.212461  1453     0   5.006  184.00
```

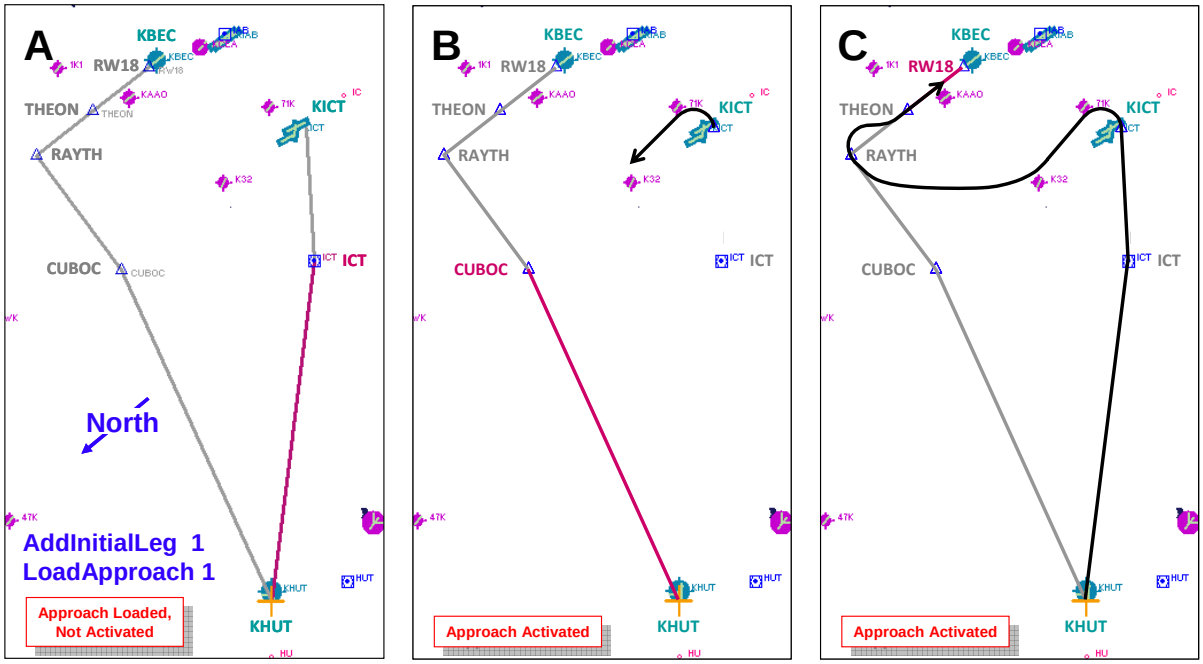
The Flight Plan and Approach segments are listed above.

In the absence of an approach segment between the aircraft and the Transition Waypoint, the aircraft flies directly to the Transition Waypoint.

Figure **B** shows the first approach segment after the approach is (automatically) activated. No Initial Leg has been added, and the aircraft proceeds directly to the Transition Waypoint, CUBOC.

Figure **C** shows the complete flight path.

Example 5.2 NewApproachAddInitialLeg = 1, LoadApproach = 1



The xml:

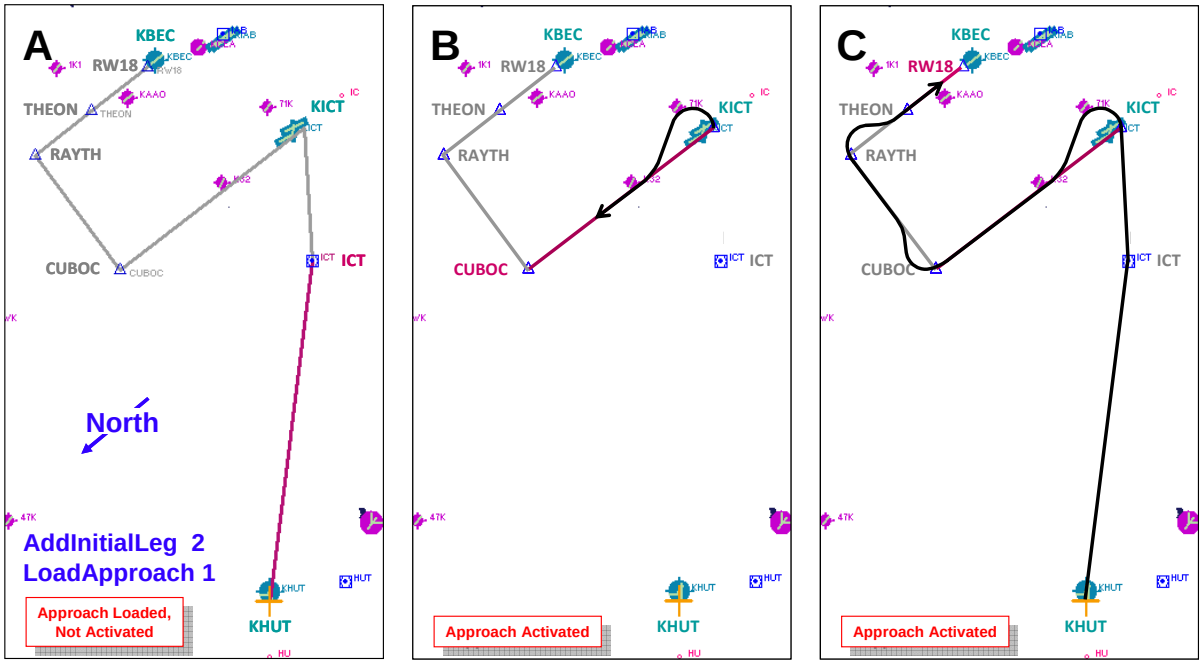
```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
1 (>@c:FlightPlanNewApproachAddInitialLeg)
1 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows that an initial approach leg from the aircraft position to the Transition Waypoint has been added. Because the approach has not been activated, it is an inactive approach segment (gray color) and the aircraft will fly towards ICT VOR-DME according to the Flight Plan, which is active.

Figure **B** shows the first approach segment after the approach is (automatically) activated. Now, the added leg (original aircraft location to Transition Waypoint) is active and the aircraft flies to intercept that segment. It is not flying directly to CUBOC, the Transition Waypoint, rather, it is flying to intercept the active approach segment. The aircraft is actually a little ahead of the waypoint, and because of the intercept algorithm, it never reaches CUBOC before the next approach segment becomes active and the aircraft turns to intercept that segment. This is admittedly an unrealistic scenario in that InitialLeg = 1 was selected but the aircraft was allowed to continue flying the Flight Plan rather than the Approach.

Figure **C** shows the complete flight path.

Example 5.3 NewApproachAddInitialLeg = 2, LoadApproach = 1



The xml:

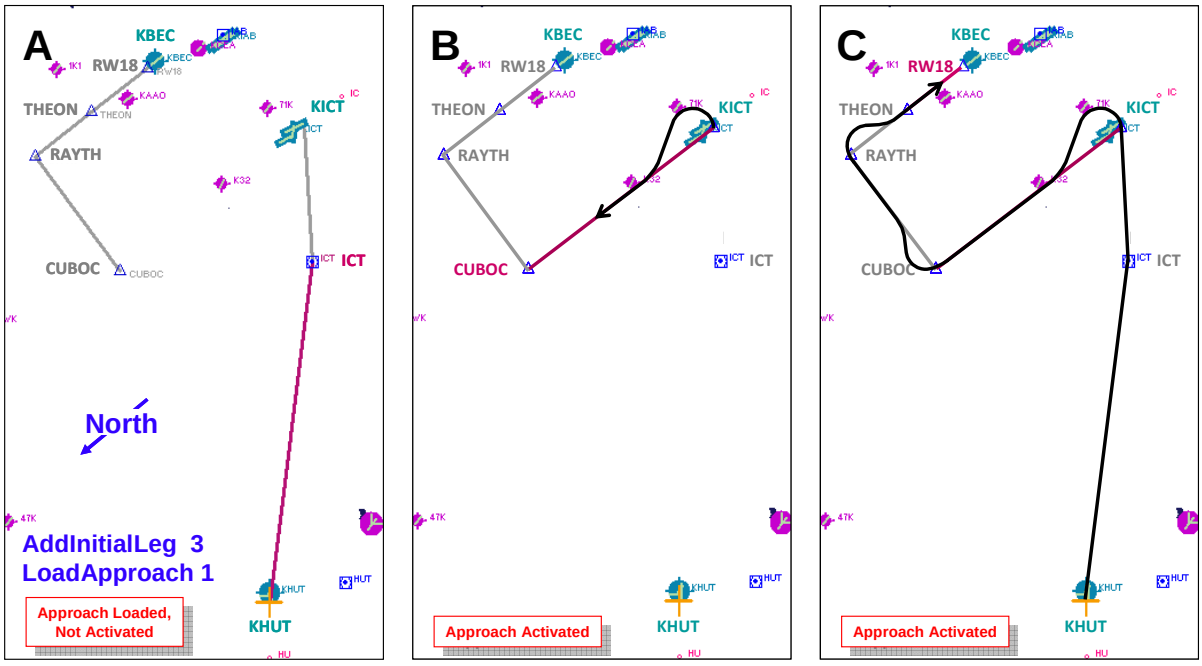
```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
2 (>@c:FlightPlanNewApproachAddInitialLeg)
1 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows that an initial approach leg from the Termination Point of the Flight Plan to the Transition Waypoint has been added.

Figure **B** shows the first approach segment after the approach is (automatically) activated. The added intitial leg is now active, and the aircraft turns to intercept that segment.

Figure **C** shows the complete flight path.

Example 5.4 NewApproachAddInitialLeg = 3, LoadApproach = 1



The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
3 (>@c:FlightPlanNewApproachAddInitialLeg)
1 (>@c:FlightPlanLoadApproach)
```

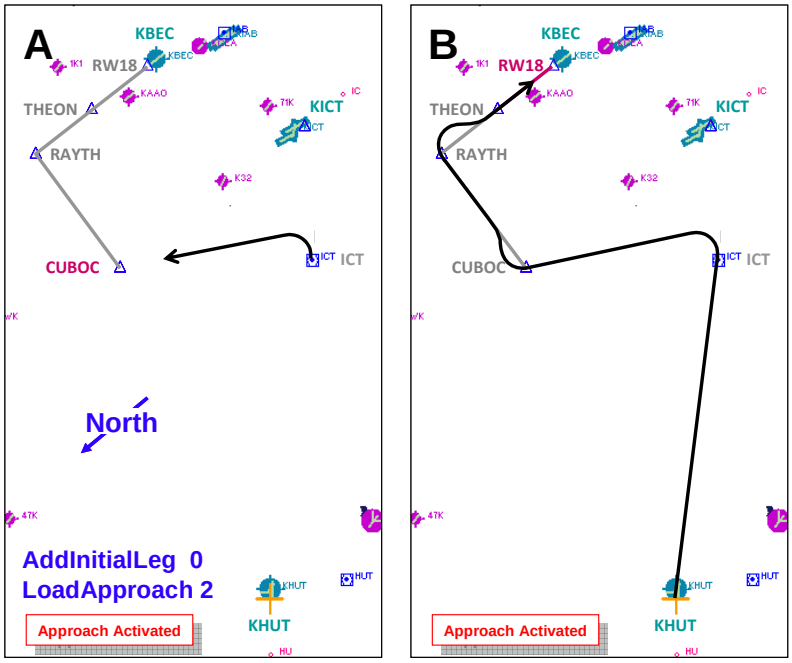
Figure **A** shows approach segments beginning at CUBOC and ending at the destination runway waypoint. No initial legs are shown.

Figure **B** shows the first approach segment after the approach is (automatically) activated. An initial approach leg from the Termination Point of the Flight Plan to the Transition Waypoint was automatically added when the approach was activated, and the aircraft turns to intercept that segment.

Figure **C** shows the complete flight path.

The next series demonstrates loading **and activating** the KBEC RNAV18 Approach in flight. The aircraft begins under control of the Flight Plan until it reaches ICT VOR-DME, at which point, the Approach is **loaded and activated**.

Example 5.5 NewApproachAddInitialLeg = 0, LoadApproach = 2



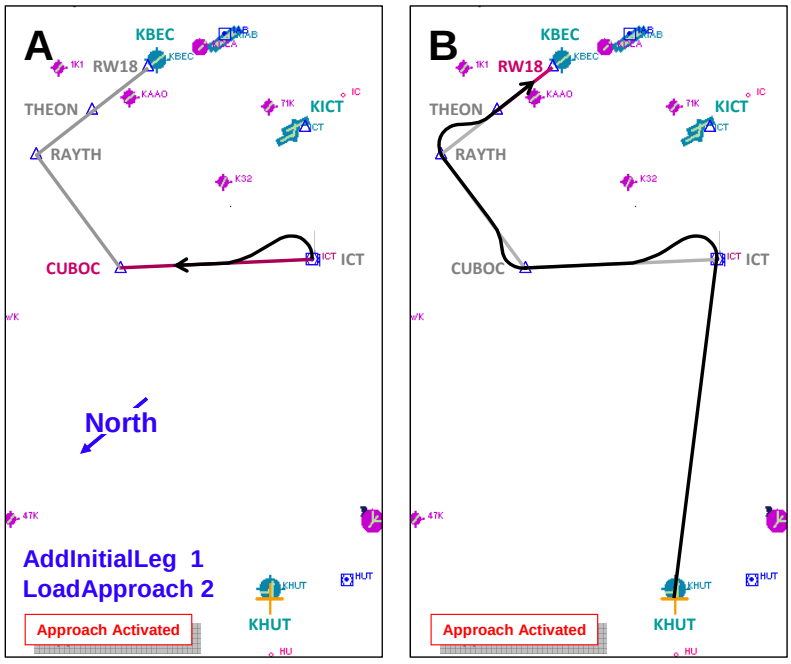
The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
0 (>@c:FlightPlanNewApproachAddInitialLeg)
2 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows the first approach segment after the approach is activated. There is no leg connecting the aircraft and the Transition Waypoint. In the absence of an approach segment between the aircraft and the Transition Waypoint, the aircraft flies directly to the waypoint.

Figure **B** shows the complete flight path.

Example 5.6 NewApproachAddInitialLeg = 1, LoadApproach = 2



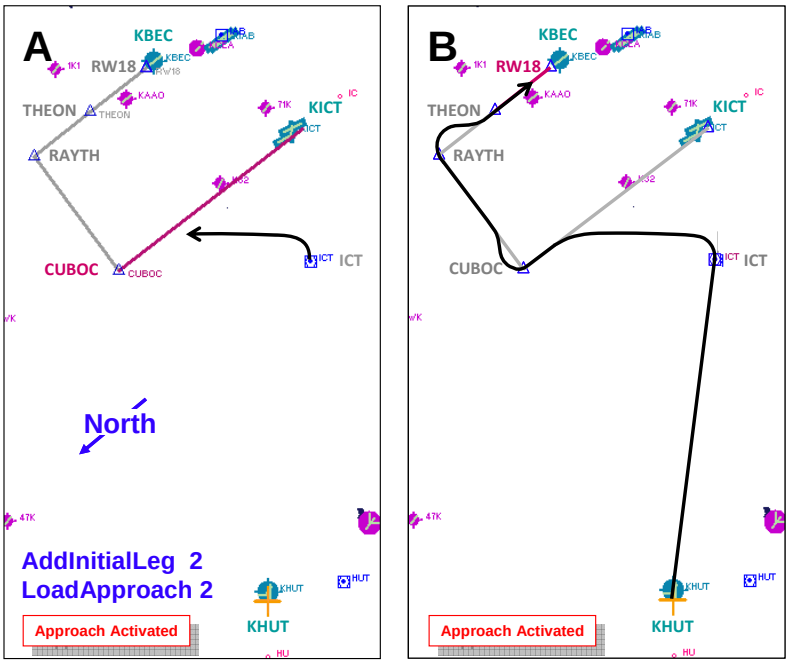
The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
1 (>@c:FlightPlanNewApproachAddInitialLeg)
2 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows the first approach segment after the approach is activated. AddInitialLeg = 1 resulted in a new segment added between the aircraft location and the Transition Waypoint. The aircraft turns to intercept the new segment, it does not fly directly to the Transition Waypoint.

Figure **B** shows the complete flight path.

Example 5.7 NewApproachAddInitialLeg = 2, LoadApproach = 2



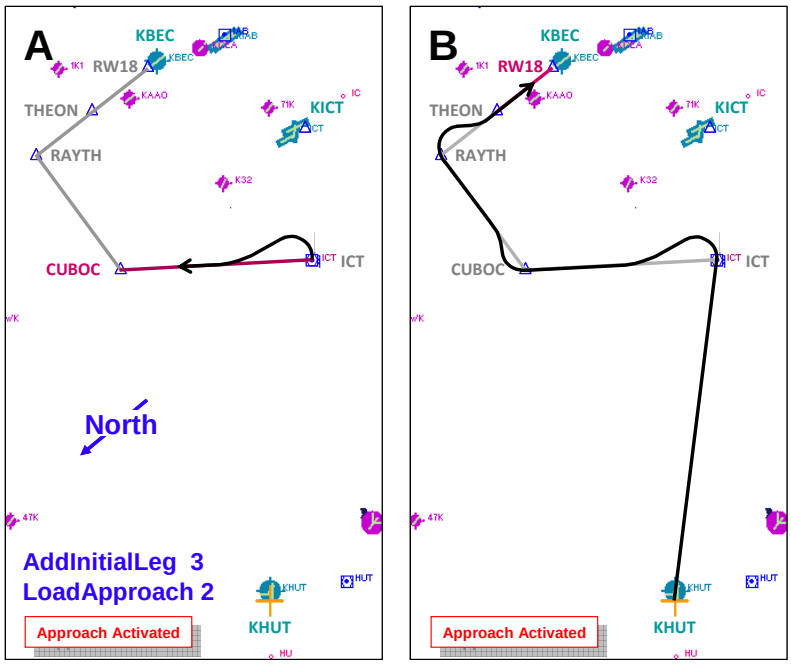
The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
2 (>@c:FlightPlanNewApproachAddInitialLeg)
2 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows the first approach segment after the approach is activated. AddInitialLeg = 2 resulted in a new segment added between the Termination Point of the Flight Plan and the Transition Waypoint. The aircraft turns to intercept the new segment; it does not fly directly to the Transition Waypoint.

Figure **B** shows the complete flight path.

Example 5.8 NewApproachAddInitialLeg = 3, LoadApproach = 2



The xml:

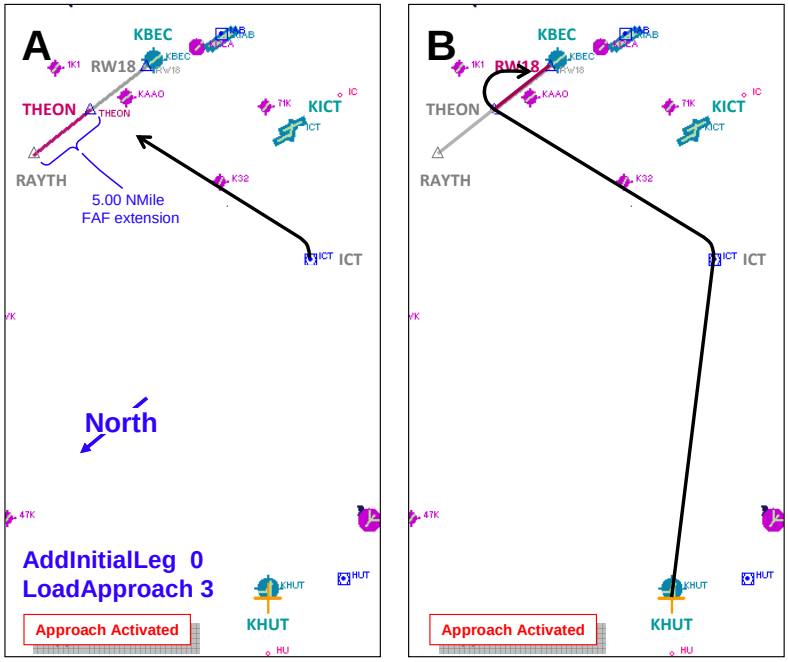
```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
5 (>@c:FlightPlanNewApproachTransition)
0 (>@c:FlightPlanNewApproachMissed)
3 (>@c:FlightPlanNewApproachAddInitialLeg)
2 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows the first approach segment after the approach is activated. This case is the same as AddInitialLeg = 1 because the Approach was activated at the same time it was loaded.

Figure **B** shows the complete flight path.

Finally, the last example demonstrates FlightPlanLoadApproach = 3, the Activate Vectors-To-Final instruction.

Example 5.9 NewApproachAddInitialLeg = 0, LoadApproach = 3



The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
1 (>@c:FlightPlanLoadApproach)
```

Vectors-To-Final replaces the existing transition with a Vectors transition. It requires that at least an approach (but not necessarily a transition, too) first be loaded or activated, or nothing will happen. If NewApproachMissed and NewApproachAddInitialLeg values have previously been entered, they will be used again. If not, the default values of zero will be used for AddInitialLeg, and 1 for Missed. In Example 5.9, the KBEC RNAV18 Approach, CUBOC Transition is already loaded when Vectors-To-Final (LoadApproach = 3) is executed.

The xml (placed following the `1 (>@c:FlightPlanLoadApproach)` statement above:

```
0 (>@c:FlightPlanNewApproachAddInitialLeg)
3 (>@c:FlightPlanLoadApproach)
```

The Flight Plan and Approach segments are listed below:

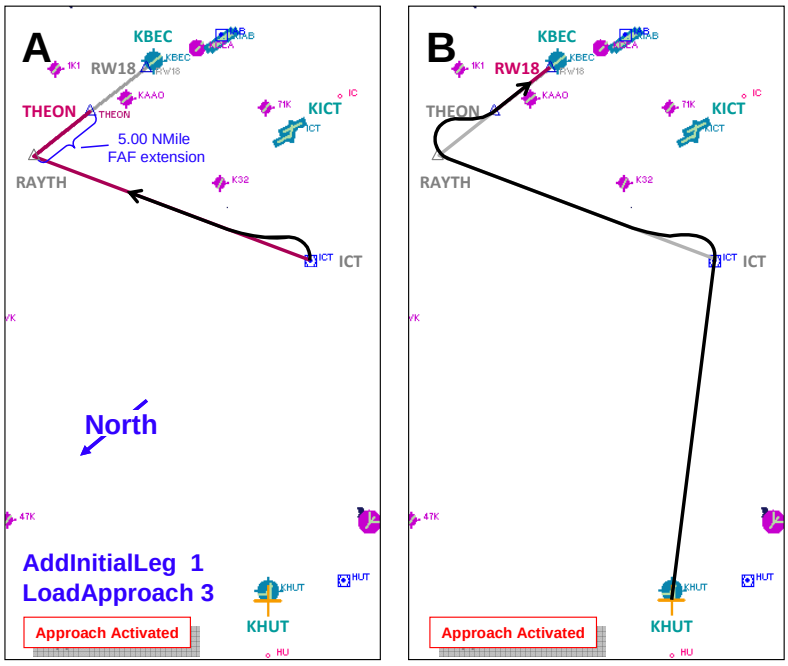
```
FLIGHT PLAN

FlightPlanTitle: KHUT to KICT
       3 :FlightPlanWaypointsNumber        2 :FlightPlanActiveWaypoint     0 :FlightPlanIsActiveWaypointLocked
       1 :FlightPlanRouteType              2 :FlightPlanFlightPlanType     1 :FlightPlanIsActiveWaypoint

------------------------------------------- FlightPlanWaypoint -------------------------------------------
     ICAO     111                    Mag                                    Dist     Dist     Rem
Idx  123456789012  Ident   Alt Type  Hdg       Lat        Lon      Dist      Ttl      Rem     Dist      ETE
0    A       KHUT    KHUT  1541    1    0   38.07383  -97.87067     0.00      0.0     33.2      0.0     0.00
1    VK3     ICT      ICT  1470    3  138   37.74517  -97.58383    23.95     23.9      9.3      0.0     6.82
2    A       KICT    KICT  1332    1  128   37.63550  -97.44583     9.29     33.2      0.0      9.4     2.60

FLIGHT PLAN NEW APPROACH: KBEC

       2 :ApprWaypointsNumber              6 :FlightPlanApprType           1 :FlightPlanWaypointApproachIndex
 RNAV 18 :FlightPlanApprName          VECTORS :FlightPlanApprTransName      0 :FlightPlanActiveApproachWaypoint
       1 :FlightPlanIsActiveApproach       2 :FlightPlanApproachIndex      0 :FlightPlanApproachTransitionIndex

----------------------------------------- FlightPlanWaypointApproach -----------------------------------------
     ICAO     111                    Latitude     Longitude   Latitude   Longitude                Leg  Course
Idx  123456789012   Name Type Mode   (Deg Min)     (Deg Min)  (Degrees)  (Degrees)   Alt  Trgt   Dist   (mag)
0    WK3KBECTHEON  THEON   11    2  37 47.2380   -97 11.5962  37.787300 -97.193269  3000     0  5.001  181.53
1    RK3KBECRW18    RW18    1    2  37 42.3165   -97 12.7477  37.705275 -97.212461  1453     0  5.006  184.00
```

Figure **A** shows the first approach segment after the Vectors-To-Final approach is loaded/activated. No initial leg is added to the approach, so the aircraft proceeds directly to the Vectors-To-Final Waypoint, the Final Approach Fix (THEON).

Note that the aircraft does <u>not</u> proceed to new Transition Waypoint which is the outboard end of the extended approach (near RAYTH). Flying direct to the Final Approach Fix results in an unacceptable, un-stabilized approach; an aircraft should never make a significant turn after reaching the Final Approach Fix. For this reason, an initial leg should be added when selecting FlightPlanLoadApproach = 3 Vectors-To-Final, as demonstrated in the next example.

Figure **B** shows the complete flight path.

Example 5.10 NewApproachAddInitialLeg = 1, LoadApproach = 3



The xml:

```
'A      KBEC' (>@c:FlightPlanNewApproachAirport)
2 (>@c:FlightPlanNewApproachApproach)
1 (>@c:FlightPlanLoadApproach)
1 (>@c:FlightPlanNewApproachAddInitialLeg)
3 (>@c:FlightPlanLoadApproach)
```

Figure **A** shows the first approach segment after the Vectors-To-Final approach is loaded/activated. In this case, an initial leg from the aircraft location to the new Transition Waypoint near RAYTH has been added. The aircraft turns to intercept that approach segment.

Figure **B** shows the complete flight path.

Note: Throughout Example 5, the flight paths depict an exaggerated turn radius in order to more clearly demonstrate the action of the different FlightPlanNewApproach selections.

En Route Navigation

- **FlightPlanWaypointIndex (enum) [Get, Set]**

The currently indexed waypoint.

- **FlightPlanWaypointLatitude**
- **FlightPlanWaypointLongitude (degrees, radians) [Get]**

Latitude and longitude of the currently indexed Waypoint. Units are degrees (decimal format, not deg, min, sec) or radians.

- **FlightPlanWaypointAltitude (feet) [Get]**

Ground elevation, or altitude, (asl) of the currently indexed waypoint. Only Waypoint types 1, 3, and 4 (Airport, VOR, NDB) have altitudes. Intersection Waypoints do not.

- **FlightPlanWaypointICAO (string) [Get]**

The ICAO of the currently indexed Waypoint.

- **FlightPlanWaypointIdent (string) [Get]**

The Ident of the currently indexed Waypoint.

- **FlightPlanWaypointAirwayIdent (string) [Get]**

The Low Altitude (Victor) or High Altitude (Jet) Airway Ident if the currently indexed flight plan leg is part of an Airway and FlightPlanRouteType = 2 or 3 (Low Altitude or High Altitude Airways).

- **FlightPlanWaypointType (enum) [Get]**

Flight Plan Waypoint Type

Bit	Name and Type #	Bit	Name and Type #	Bit	Name and Type #
0	NONE = 0	3	VOR = 3	5	USER = 5
1	AIRPORT = 1	4	NDB = 4	6	ATC = 6
2	INTERSECTION = 2				

http://msdn.microsoft.com/en-us/library/cc526954.aspx#ATC_WAYPOINT_TYPE

Waypoints added by the user through FlightPlanAddWaypoint that do not correspond to Waypoint Types 1 through 4, that is, the added Waypoint is simply a point on the map, then Waypoint Type = 5 is assigned by fs9gps.

❑ FlightPlanWaypointMinAltitude (feet) [Get]

FlightPlanWaypointMinAltitude is the Minimum En route Altitude assigned to Low Altitude Victor and High Altitude Jet Airways in the fs9gps database. Only flight plan legs that are part of a Victor or Jet Airway have a WaypointMinAltitude. Additionally, WaypointMinAltitude is returned only when FlightPlanRouteType = 2 or 3.

MEAs typically vary along an Airway, so the MEA belonging to the portion of the Airway which is the currently indexed flight plan leg is returned as WaypointMinAltitude. For example, see the various WaypointMinAltitude associated with V134 and V591 in the figure below. Non-Airway legs such as Airport KLIC to VOR FQF in the figure below will return a zero value for WaypointMinAltitude.

FLIGHT PLAN WAYPOINT

KLIC to KDTA :FlightPlanTitle
18 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
2 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 18000 :FlightPlanCruisingAltitude

FlightPlanWaypoint

Idx	ICAO 111	Ident	Airwy Ident	Alt	Min Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A KLIC	KLIC		5365	0	1	0	200	0.00	0.0	430.8	0.0	0.00	0.00	0.00	0.00	242.0	0.0	0.0
1	VK2 FQF	FQF		0	0	3	290	200	51.04	51.0	379.7	51.0	15.30	0.00	0.00	0.00	0.0	11.2	0.0
2	WK2 BREWS	BREWS	V134	0	16500	2	254	200	25.95	77.0	353.8	25.9	7.77	0.00	0.00	0.00	0.0	5.7	0.0
3	WK2 FUNDS	FUNDS	V134	0	16500	2	254	200	41.33	118.3	312.5	41.3	12.38	0.00	0.00	0.00	0.0	9.0	0.0
4	WK2 DAVVY	DAVVY	V134	0	1600	2	245	200	13.75	132.1	298.7	13.7	4.12	0.00	0.00	0.00	0.0	3.0	0.0
5	WK2 LAWSN	LAWSN	V134	0	16000	2	245	200	5.50	137.6	293.2	5.5	1.65	0.00	0.00	0.00	0.0	1.2	0.0
6	WK2 HERLS	HERLS	V134	0	16000	2	245	200	11.97	149.5	281.2	12.0	3.58	0.00	0.00	0.00	0.0	2.6	0.0
7	VK2 DBL	DBL	V134	0	14000	3	244	200	7.98	157.5	273.3	8.0	2.38	0.00	0.00	0.00	0.0	1.7	0.0
8	WK2 LINDZ	LINDZ	V134	0	14000	2	244	200	12.58	170.1	260.7	12.6	3.77	0.00	0.00	0.00	0.0	2.7	0.0
9	WK2 GLENO	GLENO	V591	0	14000	2	244	200	10.09	180.2	250.6	10.1	3.02	0.00	0.00	0.00	0.0	2.2	0.0
10	WK2 SLOLM	SLOLM	V591	0	14000	2	243	200	12.40	192.6	238.2	12.4	3.72	0.00	0.00	0.00	0.0	2.7	0.0
11	WK2 EDUKY	EDUKY	V591	0	13000	2	243	200	23.03	215.6	215.2	23.0	6.90	0.00	0.00	0.00	0.0	5.0	0.0
12	WK2 PACES	PACES	V591	0	13000	2	243	200	8.23	223.8	206.9	8.2	2.47	0.00	0.00	0.00	0.0	1.8	0.0
13	VK2 JNC	JNC	V591	0	9000	3	243	200	24.76	248.6	182.2	24.8	7.42	0.00	0.00	0.00	0.0	5.4	0.0
14	WK2 EGEZE	EGEZE	V134	0	11900	2	278	200	81.72	330.3	100.4	81.7	24.50	0.00	0.00	0.00	0.0	17.9	0.0
15	VK2 PUC	PUC	V134	0	11900	3	276	200	14.96	345.3	85.5	15.0	4.48	0.00	0.00	0.00	0.0	3.3	0.0
16	WK2 ARBIH	ARBIH	V134	0	13000	2	293	200	9.98	355.3	75.5	10.0	2.98	0.00	0.00	0.00	0.0	2.2	0.0
17	A KDTA	KDTA		4755	0	1	242	200	75.50	430.8	0.0	75.5	22.63	0.00	0.00	0.00	0.0	16.5	0.0

FLIGHT PLAN WAYPOINT

KLIC to KDTA :FlightPlanTitle
6 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
3 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 34000 :FlightPlanCruisingAltitude

FlightPlanWaypoint

Idx	ICAO 111	Ident	Airwy Ident	Alt	Min Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Rem Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A KLIC	KLIC		5365	0	1	0	200	0.00	0.0	420.4	0.0	0.00	0.00	0.00	0.00	242.0	0.0	0.0
1	VK2 FQF	FQF		0	0	3	290	200	51.04	51.0	369.4	51.0	15.30	0.00	0.00	0.00	0.0	11.2	0.0
2	VK2 EKR	EKR	J116	0	20000	3	268	200	153.81	204.8	215.5	153.8	46.13	0.00	0.00	0.00	0.0	33.7	0.0
3	WK2 HELPR	HELPR	J199	0	33000	2	249	200	122.20	327.0	93.3	122.2	36.65	0.00	0.00	0.00	0.0	26.8	0.0
4	WK2 OFFIG	OFFIG	J199	0	33000	2	247	200	41.31	368.4	52.0	41.3	12.38	0.00	0.00	0.00	0.0	9.0	0.0
5	A KDTA	KDTA		4755	0	1	239	200	52.03	420.4	0.0	52.0	15.60	0.00	0.00	0.00	0.0	11.4	0.0

There is one peculiarity to be aware of. If you want WaypointMinAltitude in **feet**, you must either omit units or specify units as 'meters'. For example, V10 airway between Hutchinson VOR intersection (Kansas, USA) and STAFF intersection (Kansas, USA) has a MEA of 3700 feet:

```
(C:fs9gps:FlightPlanWaypointMinAltitude) = 3700
```

or,

```
(C:fs9gps:FlightPlanWaypointMinAltitude, meters) = 3700
```

But, if you specify units = feet:

```
(C:fs9gps:FlightPlanWaypointMinAltitude, feet) = 12140
```

the database altitude apparently will be internally interpreted as *meters*, then multiplied by 3.281, resulting in 12,140 -- which is not the correct value. If you want meters units, then you must do the conversion manually:

```
(C:fs9gps:FlightPlanWaypointMinAltitude) 3.281 /
(>L:FlightPlanWaypointMinAltitude, meters)
```

As an additional note, not all segments of all airways within the fs9gps database have assigned MEAs. Isolated segments of many airways simply have a zero, or another obviously incorrect value while the adjoining segments may have the proper MEA.

❑ FlightPlanWaypointFrequency (MHz) [Get]

FlightPlanWaypointFrequency is not implemented in fs9gps according to an old blog by MSFT's Susan Ashcroft (when ACE's was still around). However, WaypointFrequency does return a value. It is 100.00 for all waypoint types except VORs. For VORs, it is a value that is about twice the actual VOR frequency, but it does not make sense, nor is it predictable.

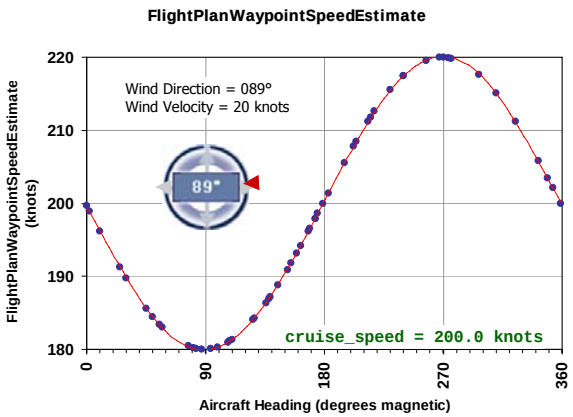
❑ FlightPlanWaypointMagneticHeading (degrees) [Get]

FlightPlanWaypointMagneticHeading is the magnetic bearing to the currently indexed Waypoint from the previous Waypoint.

WaypointMagneticHeading is not the same as FlightPlanWaypointApproachCourse. Both return magnetic bearing but WaypointMagneticHeading applies to en route Waypoints and WaypointApproachCourse applies to approach segments.

❑ FlightPlanWaypointSpeedEstimate (knots) [Get]

FlightPlanWaypointSpeedEstimate is a ground speed estimate that fs9gps derives from the aircraft's cruising airspeed defined in the aircraft.cfg file. It is used to calculate WaypointETE. WaypointETE is not updated using actual groundspeed as the flight progresses. As demonstrated in the figure, wind speed and direction affect WaypointSpeedEstimate. Note if the wind changes in-flight, SpeedEstimate and WaypointETE will not change.



❑ FlightPlanWaypointDistance (nmiles) [Get]

FlightPlanWaypointDistance is the length of the currently indexed Flight Plan leg.

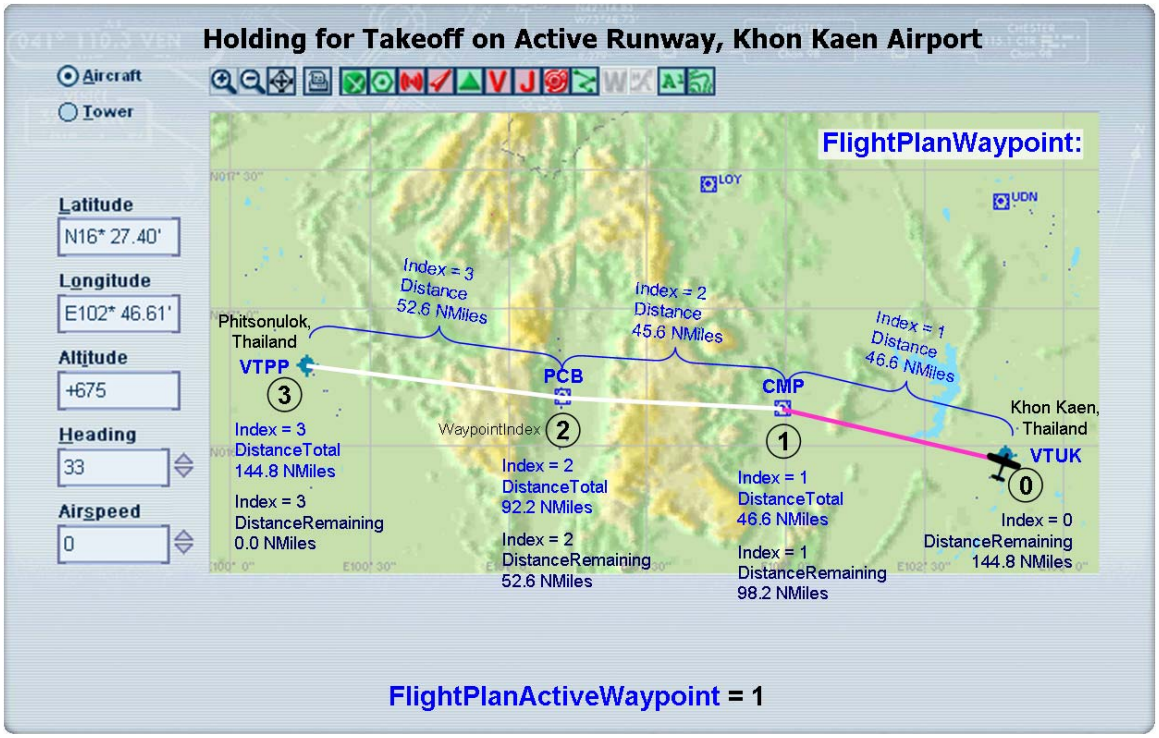
❑ FlightPlanWaypointDistanceTotal (nmiles) [Get]

FlightPlanWaypointDistanceTotal is the cumulative distance of all Flight Plan legs starting at Waypoint index zero (the departure airport) through the currently indexed Waypoint. When the index points to the last Waypoint (the destination airport), DistanceTotal is the total length of the flight plan measured along flight legs.

❑ FlightPlanWaypointDistanceRemaining (nmiles) [Get]

FlightPlanWaypointDistanceRemaining is the distance from the currently indexed Waypoint to the last Waypoint, the destination airport.

FlightPlanWaypointDistance, DistanceTotal, DistanceRemaining



FLIGHT PLAN WAYPOINT

VTUK to VTPP :FlightPlanTitle
4 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
0 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 8000 :FlightPlanCruisingAltitude

FlightPlanWaypoint

Idx	ICAO 111 123456789012	Ident	Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Dist Rem	Rem Dist	Rem Ttl Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A VTUK	VTUK	670	1	0	200	0.00	0.0	144.8	0.0	0.0	0.00	0.00	0.00	0.00	242.0	0.0	0.0
1	VVT CMP	CMP	650	3	284	200	46.60	46.6	98.2	46.6	46.6	13.97	0.00	0.00	0.00	0.0	10.2	0.0
2	VVT PCB	PCB	449	3	274	200	45.58	92.2	52.6	45.6	92.2	13.67	0.00	0.00	0.00	0.0	10.0	0.0
3	A VTPP	VTPP	145	1	277	200	52.59	144.8	0.0	52.6	144.8	15.77	0.00	0.00	0.00	0.0	11.5	0.0

❑ FlightPlanWaypointRemainingDistance (nmiles) [Get]

FlightPlanWaypointRemainingDistance is the leg distance remaining to be flown. For the active waypoint, that is, for the segment currently being flown, it is the remaining distance from the aircraft's current position to the next Waypoint. For Waypoints beyond that, it is just the total length of that leg. For Waypoints already passed, WaypointRemainingDistance is 0.0.

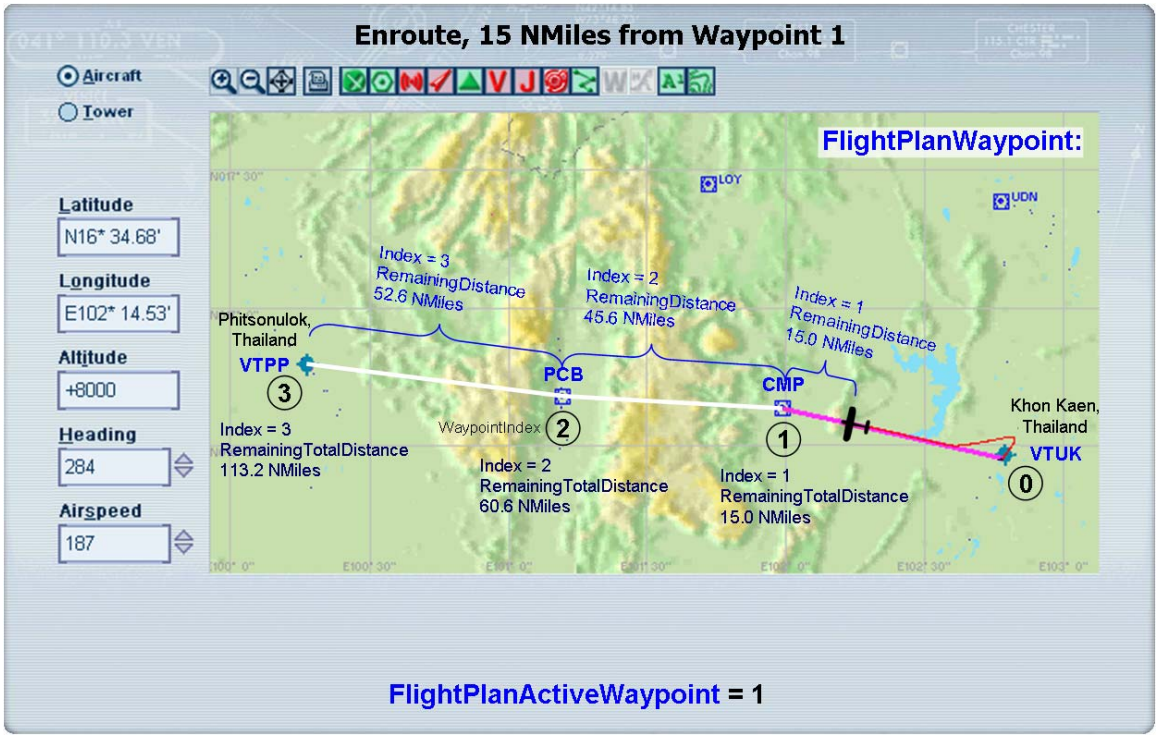
In the figure below, the aircraft is en route, and 15.0 NMiles remain before reaching Waypoint 1, CMP VOR. FlightPlanActiveWaypoint = 1. WaypointRemainingDistance for Index 1 is therefore, 15.0. Because the aircraft is not yet on leg 2 (ActiveWaypoint = 2) or leg 3 (ActiveWaypoint = 3), WaypointRemainingDistance for those Waypoints is still the total length of the respective Flight Plan leg.

WaypointRemainingDistance for the ActiveWaypoint counts down as the flight progresses.

❑ FlightPlanWaypointRemainingTotalDistance (nmiles) [Get]

FlightPlanWaypointRemainingTotalDistance is the cumulative remaining distance from current aircraft position to the indexed waypoint. It is the same as RemainingDistance when the currently indexed waypoint is the Active Waypoint. When the indexed waypoint is the destination airport, RemainingDistance represents the total distance remaining in the flight. RemainingDistance is measured along the flight path; it is not a direct-to measurement.

FlightPlanWaypointRemainingDistance, RemainingTotalDistance



FLIGHT PLAN WAYPOINT

VTUK to VTPP :FlightPlanTitle
4 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
0 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 8000 :FlightPlanCruisingAltitude

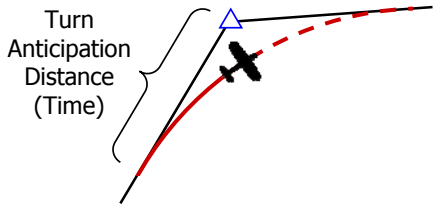
FlightPlanWaypoint

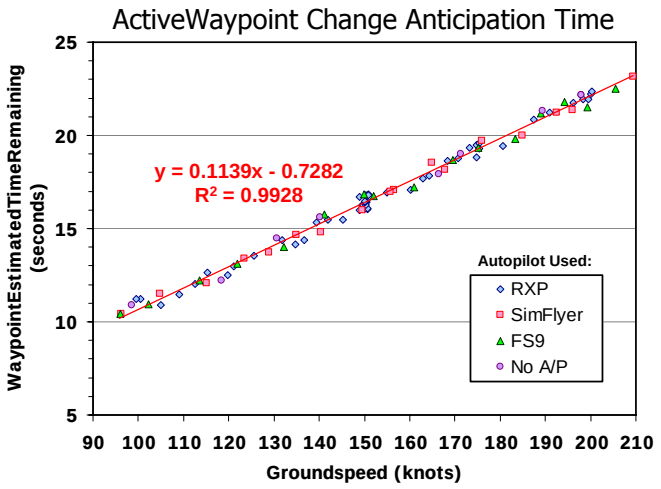
Idx	ICAO 111 1234567890112	Ident	Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Dist Rem	Rem Dist	Rem Ttl Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A	VTUK	VTUK	670	1	0	200	0.00	0.0	144.8	0.0	0.0	0.00	0.00	0.00	10.72	241.6	0.0	0.0
1	VVT	CMP	CMP	650	3	284	200	46.60	46.6	98.2	15.0	15.0	13.97	0.00	4.28	10.99	0.0	10.2	0.0
2	VVT	PCB	PCB	449	3	274	200	45.58	92.2	52.6	45.6	60.6	13.67	0.00	13.00	11.21	0.0	10.0	0.0
3	A	VTPP	VTPP	145	1	277	200	52.59	144.8	0.0	52.6	113.2	15.77	0.00	15.00	11.46	0.0	11.5	0.0

TURN ANTICIPATION

To accomplish a smooth turn to the next heading as the aircraft approaches a waypoint, the aircraft must begin its turn before actually reaching the waypoint. The distance at which this happens is the Turn Anticipation Distance. Turn Anticipation Distance algorithms and rules of thumb in the literature vary but most are a function of the amount of turn; how many degrees change of direction. With the fs9gps module, however, Turn Anticipation is a function of groundspeed, and the ActiveWaypoint changes, advancing by 1, when a certain seconds-to-waypoint (WaypointEstimatedTimeRemaining) time is reached. This is independent of the amount of degrees turned. Several (maybe most) popular flight sim autopilots, including the stock FS9 Bendix-King, Reality-XP STEC55X, and the Simflyer STEC55X autopilots which I have tested, initiate the turn when the ActiveWaypoint changes, or, when on approach, when FlightPlanActiveApproachWaypoint changes.

I have timed hundreds of ActiveWaypoint changes at random groundspeeds and direction changes using a variety of autopilots and also without autopilot. The conclusions are that the timing of the ActiveWaypoint change is independent of the use of an autopilot (of course) and the amount of turn, and that TimeRemaining when the ActiveWaypoint change occurs is a linear function of groundspeed (which means it's a power function of distance), as shown in the graph below. The Groundspeed vs. WaypointEstimatedTimeRemaining relationship is built into fs9gps.





The table below captures FlightPlanWaypoint variable status the moment *before* ActiveWaypoint changes from 1 to 2. The aircraft is 0.38 minutes = 24 seconds

EstimatedTimeRemaining from Waypoint 2. There are 1.40 NMiles RemainingDistance in Flight Plan leg 1.

FLIGHT PLAN WAYPOINT

VTUK to VTPP :FlightPlanTitle
4 :FlightPlanWaypointsNumber 1 :FlightPlanActiveWaypoint
0 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 8000 :FlightPlanCruisingAltitude

FlightPlanWaypoint

Idx	ICAO 111	Ident	Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Dist Rem	Rem Dist	Rem Ttl Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A VTUK	VTUK	670	1	0	200	0.00	0.0	144.8	0.0	0.0	0.00	0.00	0.00	10.72	241.6	0.0	0.0
1	VVT CMP	CMP	650	3	284	200	46.60	46.6	98.2	1.4	1.4	13.97	0.00	0.40	10.99	0.0	10.2	0.0
2	VVT PCB	PCB	449	3	274	200	45.58	92.2	52.6	45.6	47.0	13.67	0.00	13.00	11.21	0.0	10.0	0.0
3	A VTPP	VTPP	145	1	277	200	52.59	144.8	0.0	52.6	99.6	15.77	0.00	15.00	11.46	0.0	11.5	0.0

ActiveWaypoint changes to 2 when EstimatedTimeRemaining = 0.38 (23 seconds) at which point, the aircraft is 1.34 NMiles from Waypoint 2. At the instant ActiveWaypoint changes, Waypoint 1 RemainingDistance becomes 0.0 and the 1.34 NMiles is moved to Waypoint 2, as demonstrated in the table below. Additionally, Waypoint Index 1 EstimatedTimeRemaining becomes 0.0. Waypoint Index 2 EstimatedTimeRemaining increases from 13.00 to 13.37 minutes to accommodate the additional 1.3 NMiles.

ActiveWaypoint Change Anticipation Time

FLIGHT PLAN WAYPOINT

VTUK to VTPP :FlightPlanTitle
4 :FlightPlanWaypointsNumber 2 :FlightPlanActiveWaypoint
0 :FlightPlanRouteType 2 :FlightPlanFlightPlanType 8000 :FlightPlanCruisingAltitude

FlightPlanWaypoint

Idx	ICAO 111	Ident	Alt	Type	Mag Hdg	Spd Est	Dist	Dist Ttl	Dist Rem	Rem Dist	Rem Ttl Dist	ETE	ATE	Est Time Rem	ETA	Fuel Rem @ Arvl	Est Fuel Cons	Act Fuel Cons
0	A VTUK	VTUK	670	1	0	200	0.00	0.0	144.8	0.0	0.0	0.00	0.00	0.00	10.72	241.6	0.0	0.0
1	VVT CMP	CMP	650	3	284	200	46.60	46.6	98.2	0.0	0.0	13.97	15.93	0.00	10.99	228.8	10.2	12.8
2	VVT PCB	PCB	449	3	274	200	45.58	92.2	52.6	1.3	1.3	13.67	0.00	13.37	11.21	0.0	10.0	0.0
3	A VTPP	VTPP	145	1	277	200	52.59	144.8	0.0	52.6	53.9	15.77	0.00	15.00	11.46	0.0	11.5	0.0

A second later, Waypoint Index 2 RemainingDistance is updated by adding Waypoint Index 2 distance, 45.6 NMiles, which results in Waypoint 2 RemainingDistance = 46.9 NMiles.

```
FLIGHT PLAN WAYPOINT

VTUK to VTPP :FlightPlanTitle
 4 :FlightPlanWaypointsNumber    2 :FlightPlanActiveWaypoint
 0 :FlightPlanRouteType          2 :FlightPlanFlightPlanType     8000 :FlightPlanCruisingAltitude

------------------------------------------------- FlightPlanWaypoint -------------------------------------------------
                                                                                  Rem                  Est        Fuel   Est   Act
     ICAO     111                        Mag  Spd         Dist  Dist   Rem    Ttl                  Time       Rem @  Fuel  Fuel
Idx  123456789012  Ident   Alt Type  Hdg  Est   Dist    Ttl   Rem   Dist   Dist    ETE   ATE     Rem   ETA  Arvl  Cons  Cons
0    A      VTUK    VTUK   670    1    0  200   0.00    0.0 144.8    0.0    0.0   0.00  0.00    0.00 10.72 241.6   0.0   0.0
1    VVT    CMP      CMP   650    3  284  200  46.60   46.6  98.2    0.0    0.0  13.97 15.93    0.00 10.99 228.8  10.2  12.8
2    VVT    PCB      PCB   449    3  274  200  45.58   92.2  52.6   46.9   46.9  13.67  0.00   13.37 11.21   0.0  10.0   0.0
3    A      VTPP    VTPP   145    1  277  200  52.59  144.8   0.0   52.6   99.5  15.77  0.00   15.00 11.46   0.0  11.5   0.0
```

It's tedious to go through the steps of a waypoint change like this, but it's important to understand because Turn Anticipation, or, more to the point, Waypoint Change Anticipation affects all En Route Waypoints and Approach Segments and Sub-Segments in fs9gps.

❑ FlightPlanWaypointTimeZoneDeviation (minutes) [Get]

Because ETA is in Local Time, Time Zone Deviation is necessary to adjust to Local Time for flights that cross time zone boundaries.

❑ FlightPlanWaypointETE (minutes) [Get]

FlightPlanWaypointETE is the estimated en route time to the currently indexed waypoint. This estimate is calculated using FlightPlanWaypointSpeedEstimate which in turn, is derived from the aircraft cruising airspeed found in the aircraft.cfg file incorporating wind speed and direction.

FlightPlanWaypointETE is established when the flight plan is loaded and does not change during flight regardless of the groundspeed, position, or heading of the aircraft.

❑ FlightPlanWaypointATE (minutes) [Get]

FlightPlanWaypointATE is the actual elapsed time from ActiveWaypoint change to ActiveWaypoint change. Before reaching the next Waypoint, or, more precisely, before the ActiveWaypoint changes, WaypointATE returns zeros.

❑ FlightPlanWaypointEstimatedTimeRemaining (minutes) [Get]

FlightPlanWaypointEstimatedTimeRemaining is the time remaining until the currently indexed waypoint is reached. It is calculated by dividing WaypointDistanceRemaining by the current aircraft ground speed (A:GROUND VELOCITY). EstimatedTimeRemaining is associated with individual flight plan legs and is not cumulative involving multiple legs.

For the active waypoint, that is, for the flight plan leg currently being flown, EstimatedTimeRemaining is the remaining time from the aircraft's current position to the

next Waypoint, so it counts down as the aircraft flies toward that Waypoint. For Waypoints beyond that, it is the time required to fly the total length of that leg at the current groundspeed. For Waypoints already passed, EstimatedTimeRemaining is 0.0.

❑ FlightPlanWaypointETA (hours) [Get]

FlightPlanWaypointETA is the estimated time of arrival at the currently indexed Waypoint. It is a Local Time reference, so the most sensible units are probably Hours. Some points to consider:

- ❑ WaypointETA for the currently indexed Waypoint is calculated by adding WaypointEstimatedTimeRemaining for the currently indexed Waypoint to Local Time (e.g., EstimatedTimeRemaining + `E:LOCAL TIME`).

- ❑ There is a slightly different rule, however, that applies to WaypointIndex 0. WaypointETA for WaypointIndex 0, the departure airport, is 0.00 while the aircraft is on the ground. WaypointETA is set to `E:LOCAL TIME` at the moment the aircraft becomes airborne, when `A:SIM ON GROUND, bool` = 0. It does not matter where the aircraft starts its flight plan: at a Parking Gate, on the Active Runway, lots of taxiing or little taxiing, WaypointETA for WaypointIndex 0 is 0.00 until takeoff.

- ❑ When the aircraft passes a Waypoint (or, being precise, when ActiveWaypoint changes), WaypointETA equals Local Time, and it does not change after that regardless of the groundspeed, position, or heading of the aircraft because EstimatedTimeRemaining for a Waypoint that has been passed is zero. In other words, WaypointETA becomes Waypoint Actual Time of Arrival when a Waypoint is passed.

❑ FlightPlanWaypointFuelRemainedAtArrival (gallons) [Get]

Fuel calculations are based on fuel consumption rates derived from the aircraft model design and configuration and fuel quantity values. Fuel consumption rates and quantity can be accessed from A:Vars, for example, `(A:ENG1 FUEL FLOW GPH, gallons per hour)` and `(A:FUEL TOTAL QUANTITY, gallons)`.

❑ FlightPlanWaypointEstimatedFuelConsumption (gallons) [Get]

Fuel calculations are based on fuel consumption rates derived from the aircraft model design and configuration and fuel quantity values. Fuel consumption rates and quantity can be accessed from A:Vars, for example, `(A:ENG1 FUEL FLOW GPH, gallons per hour)` and `(A:FUEL TOTAL QUANTITY, gallons)`.

❑ FlightPlanWaypointActualFuelConsumption (gallons) [Get]

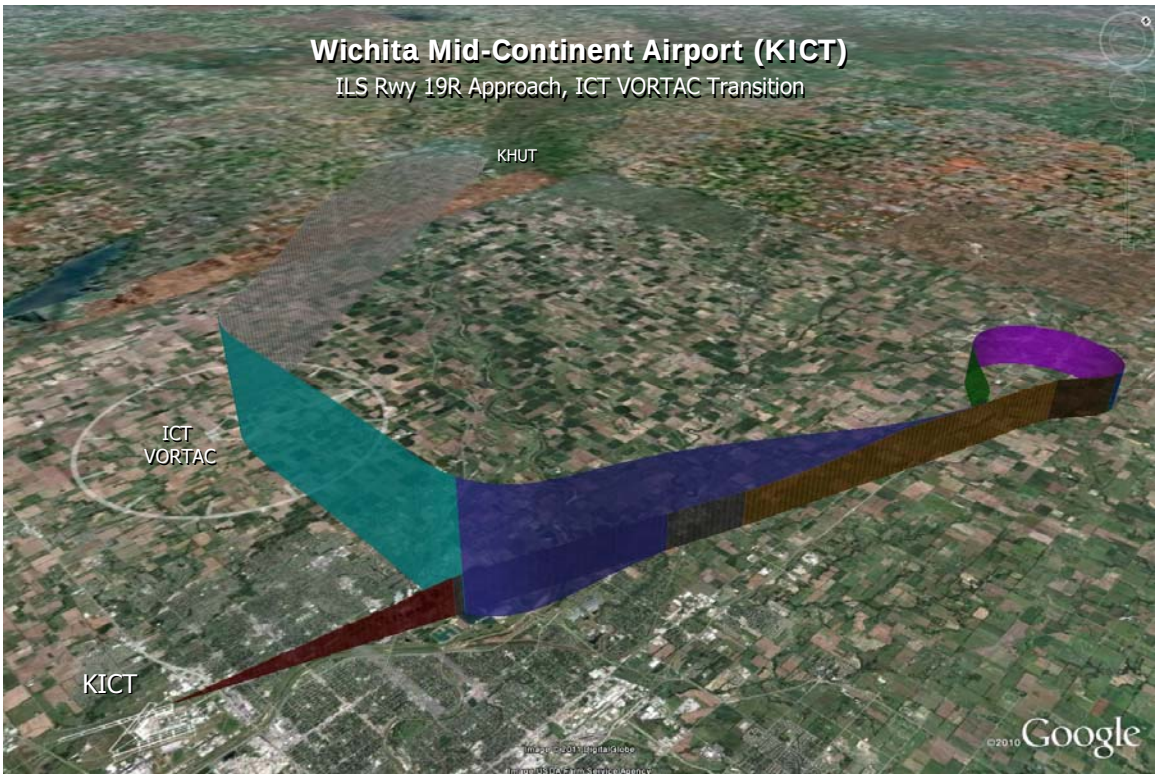
Fuel calculations are based on fuel consumption rates derived from the aircraft model design and configuration and fuel quantity values. Fuel consumption rates and quantity can be accessed from A:Vars, for example, `(A:ENG1 FUEL FLOW GPH, gallons per hour)` and `(A:FUEL TOTAL QUANTITY, gallons)`.

Instrument Approaches

Variables of the FlightPlanApproach and FlightPlanWaypointApproach groups define the flight path according to Instrument Approach Procedures from the en route approach transition point through the approach procedure, to the landing point, and finally to the missed approach procedure and holding pattern.

Throughout the discussion of Approach variables, an example instrument flight from Hutchinson Municipal Airport (Hutchinson, Kansas, USA, "KHUT") to Wichita Mid-Continent Airport (Wichita, Kansas, USA, "KICT") is used, incorporating the ILS Rwy 19R Approach, ICT VORTAC Transition into Wichita.

In the example shown below, the aircraft departs Hutchinson and proceeds to the Transition Fix, ICT VORTAC. The Approach Transition in this particular simulation is flown at 7000' altitude and from there, according to the descent profile for the approach. Between rotate and flare, the aircraft is flown by the stock FS9 Bendix-King Radio Autopilot with the user controlling altitude except on Final Approach. Approach segment and sub-segment colors match those used in the discussion of the KICT ILS Rwy 19 Approach in this section.

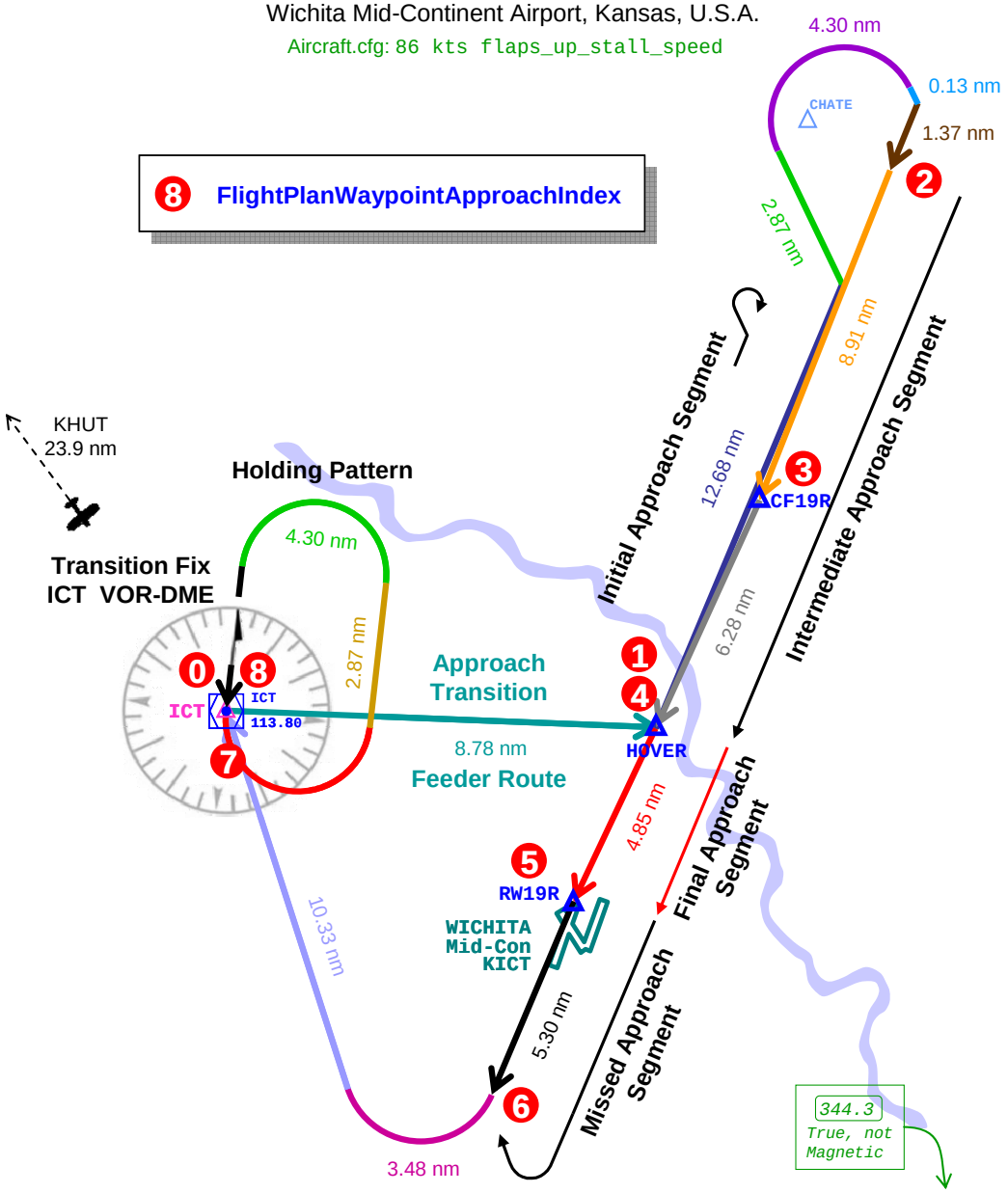


FS9 Transition and Approach Segments

KICT ILS Rwy 19R; ICT Transition

Wichita Mid-Continent Airport, Kansas, U.S.A.

Aircraft.cfg: `86 kts flaps_up_stall_speed`




```
FLIGHT PLAN WAYPOINT
KHUT to KICT :FlightPlanTitle   3000 :FltPlnCruisingAltitude
  1 :FltPlnActiveWaypoint          3 :FltPlnWaypointsNumber     1 :FltPlnRouteType     2 :FltPlnFlightPlanType
------------------------------------------ FlightPlanWaypoint ------------------------------------------

     ICAO      111                          Mag   Spd          Dist  Dist     Rem                  Est        Fuel   Est   Act
Idx  123456789012  Ident   Alt Type  Hdg   Est   Dist   Ttl   Rem   Dist   ETE   ATE   Time  ETA  Rem @ Fuel  Cons  Cons
0    A       KHUT    KHUT  1541    1    0   200   0.00   0.0   33.2   0.00  0.00  0.00  0.00  0.00 242.0  0.0   0.0
1    VK3     ICT      ICT  1470    3  138   200  23.95  23.9    9.3  23.95  7.17  0.00  0.00  0.00   0.0  4.3   0.0
2    A       KICT    KICT  1332    1  128   200   9.29  33.2    0.0   9.29  2.78  0.00  0.00  0.00   0.0  1.7   0.0

FLIGHT PLAN WAYPOINT APPROACH
 13 :FltPlnApprType   9 :ApprWptsNum   0 :ActiveApprWpt   ILS 19R :FltPlnApprName      ICT :FltPlnApprTransName
------------------------------------------ FlightPlanWaypointApproach ------------------------------------------

                                                                                  Leg    Leg                Rem
     ICAO      111                                                        Leg    Ttl    From      Rem      Ttl
Idx  123456789012  Name  Type Mode        Lat          Lon          Alt  Trgt  Dist   Dist   Dist      Dist     Dist   Course
0    VK3     ICT      ICT     1    1   37.745233  -97.583825       0     0  0.00   0.00  83.61      0.00     0.00    -1.0
1    WK3KICTHOVER  HOVER     1    1   37.737564  -97.398989    3500     0  8.78   8.78  83.61      8.78     8.78    93.1
2    WK3KICTHOVER  HOVER     3    1   37.977508  -97.297155    3500     0 21.35  30.13  74.83     21.35    30.13   328.0
3    WK3KICTCF19R  CF19R     1    2   37.835872  -97.353867    3500     0  8.91  39.04  53.48      8.91    39.04   197.5
4    WK3KICTHOVER  HOVER     1    2   37.737564  -97.398989    3000     0  6.28  45.32  44.56      6.28    45.32   193.0
5    RK3KICTRW19R  RW19R     1    2   37.661604  -97.433817    1382     0  4.85  50.17  38.29      4.85    50.17   193.0
6                            9    3   37.578144  -97.470076    3500  3500  5.30  55.46  33.44      5.30    55.46   193.0
7    VK3     ICT      ICT     1    3   37.745233  -97.583825    3500     0 13.81  69.27  28.14     13.81    69.27   344.3
8    VK3     ICT      ICT     6    3   37.745233  -97.583825    3500     0 14.34  83.61  14.34     14.34    83.61   180.0
```

KICT
WICHITA MID-CONTINENT

JEPPESEN
11-4

WICHITA, KAN
ILS Rwy 19R

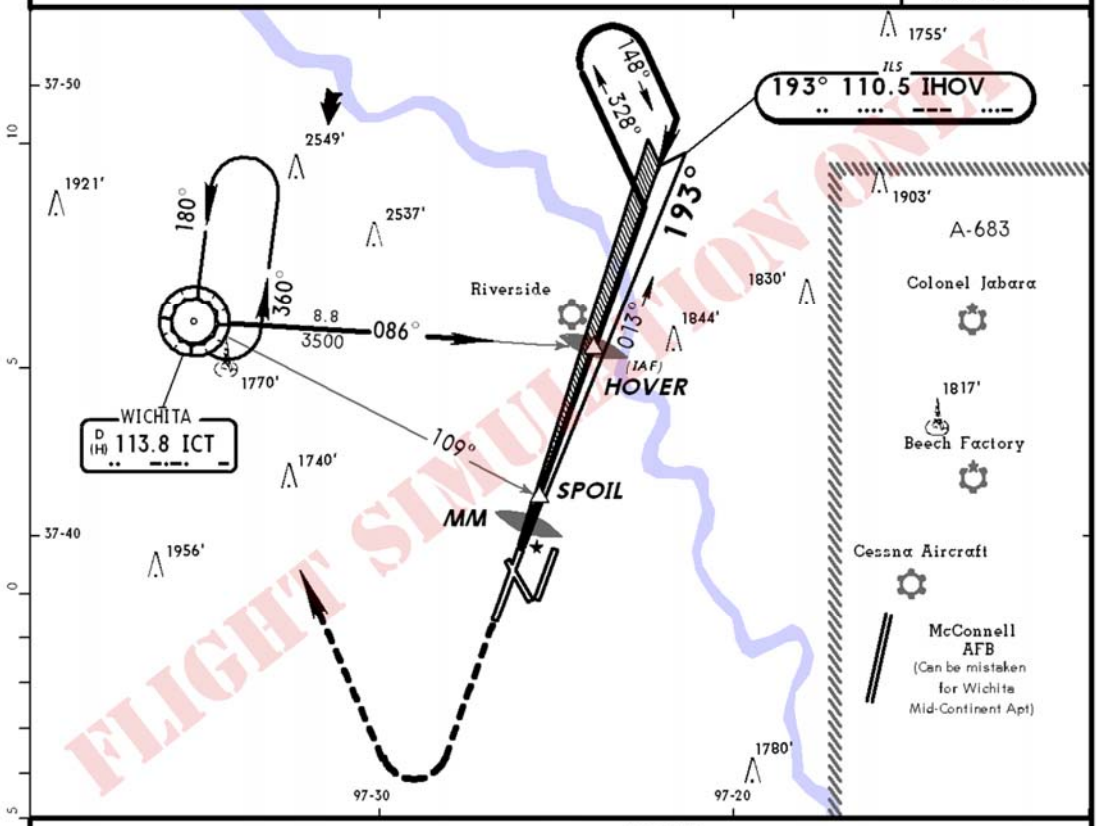
ATIS	WICHITA Approach (R)	WICHITA Tower	Ground
125.15	126.7	118.2	121.9

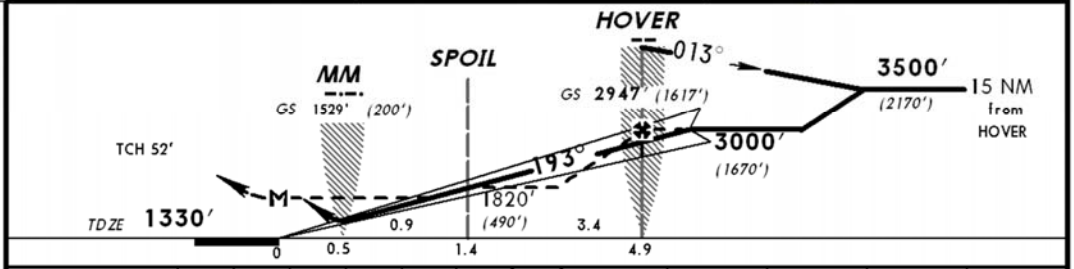
BRIEFING STRIP

LOC IHOV	Final Apch Crs	GS HOVER	ILS DA(H)	Apt Elev / TDZE
110.5	193°	2947′ (1617′)	1530′ (200′)	1333′ / 1330′

MISSED APCH: Climb to 3500′ then RIGHT turn direct ICT VOR and hold.

4100′
MSA ICT VOR





Gnd speed-Kts	70	90	100	120	140	160
GS 3.00°	377	485	539	647	754	862
HOVER to MAP 4.9	4:12	3:16	2:56	2:27	2:06	1:50

MALSR | 3500′ | RT | D | ICT 113.8

STRAIGHT-IN LANDING RWY 19R

ILS DA(H) 1530′ (200′)

	FULL	RAIL or ALS out
A, B, C	RVR 24 or 1/2	RVR 40 or 3/4
D	RVR 40 or 3/4	RVR 50 or 1

LOC (GS out)

	MDA(H) 1660′ (330′) With Spoil			MDA(H) 1820′ (490′) Without Spoil		
		RAIL out	ALS out		RAIL out	ALS out
A, B	RVR 24 or 1/2	RVR 40 or 3/4	RVR 50 or 1	RVR 24 or 1/2	RVR 40 or 3/4	RVR 50 or 1
C				RVR 40 or 3/4	RVR 60 or 1 1/4	
D	RVR 40 or 3/4	RVR 50 or 1		RVR 50 or 1	1 1/2	

CIRCLE-TO-LAND

Max Kts	With Spoil MDA(H)	Without Spoil MDA(H)
90, 120	1800′ (467′) -1	1820′ (487′) -1
140	1800′ (467′) -1 1/2	1820′ (487′) -1 1/2
165	1900′ (567′) -2	1900′ (567′) -2

❑ FlightPlanApproachWaypointType (enum) [Get]

FlightPlanApproachWaypointType is an enum referring to the navigation procedure objective of the currently *active* (as opposed to currently indexed) approach segment.

Approach Waypoint Type

Bit	Name and Type #	Bit	Name and Type #	Bit	Name and Type #
0	NONE = 0	4	DME_ARC_LEFT = 4	8	DISTANCE = 8
1	FIX = 1	5	DME_ARC_RIGHT = 5	9	ALTITUDE = 9
2	PROC_TURN_LEFT = 2	6	HOLDING_LEFT = 6	10	MANUAL_SEQ = 10
3	PROC_TURN_RIGHT = 3	7	HOLDING_RIGHT = 7	11	VECTORS_TO_FINAL = 11

http://msdn.microsoft.com/en-us/library/cc526954.aspx#GPS_APPROACH_WAYPOINT_TYPE

❑ FlightPlanApproachMode (enum) [Get]

FlightPlanApproachMode is a number used to describe the *active* segment of the approach. ApproachMode has some similarity to US F.A.A. Approach Segment nomenclature as demonstrated below.

Approach Mode

Bit	Name and Type #	Bit	Name and Type #
0	NONE = 0	2	FINAL = 2
1	TRANSITION = 1	3	MISSED = 3

http://msdn.microsoft.com/en-us/library/cc526954.aspx#GPS_APPR_TYPE

F.A.A. Instrument Approach Procedure Segments	FS9 FlightPlanApproachMode		F.A.A. Instrument Approach Procedure Segments	FS9 FlightPlanApproachMode	
En Route	0	NONE	Final Approach	2	FINAL
Feeder Route	1	TRANSITION	Missed Approach	3	MISSED
Initial Approach	1	TRANSITION	Holding Pattern	3	MISSED
Intermediate Approach	2	FINAL			

❑ FlightPlanApproachSegmentType (enum) [Get]

FlightPlanApproachSegmentType is a number indicating the direction of turn within an approach segment.

- ❑ 0 = No turn
- ❑ 1 = Right turn
- ❑ 2 = Left turn

Approach Segments and Sub-Segments: Approach segments containing both turns and straight sections *within the segment* are divided into sub-segments. SegmentType is applied to the sub-segment representing the continuous turn (e.g., procedure turn, missed approach turn to holding fix, holding pattern turns).

FlightPlanApproachSegmentType is an intra-segment variable and does not apply to turns from one straight approach segment to the next such as the turn to the outbound initial approach segment that occurs when the aircraft reaches the Initial Approach Fix.

Confusing without a picture, so refer to the detailed dissection of KICT ILS 19R ICT Transition at the end of this section for additional clarification.

❑ FlightPlanApproachSegmentDistance (nmiles) [Get]

FlightPlanApproachSegmentDistance is the remaining distance within the currently indexed approach segment or sub-segment between the aircraft position and the termination point of the segment. It counts downs as the aircraft proceeds toward the segment termination point.

ApproachSegmentDistance can be used to measure and keep track of the length and remaining distance of sub-segments whereas WaypointApproachLegDistance and WaypointApproachRemainingDistance do not get into the sub-segment level.

Also handy is that ApproachSegmentDistance and ApproachSegmentLength are also active during the En Route flight phase, returning the same values as WaypointRemainingDistance and WaypointDistance, respectively.

❑ FlightPlanApproachSegmentLength (nmiles) [Get]

FlightPlanApproachSegmentLength is the total length of the indexed approach segment. When the approach segment involves turns, (e.g., a procedure turn or missed approach turn to a holding fix) ApproachSegmentLength is the total length of the active sub-segment. As discussed later on, sub-segments do not have separate index pointers.

❑ FlightPlanApproachIsWaypointRunway (bool) [Get]

Final Approach to a Runway. FlightPlanApproachIsWaypointRunway equals 1 when the active approach segment is the Final Approach segment and the termination point is a destination runway waypoint (e.g., RW18). Also, FlightPlanWaypointApproachMode = 2 (Final).

For all other approach segments, ApproachIsWaypointRunway equals 0.

❑ FlightPlanApproachAirportIdent (string) [Get]

FlightPlanApproachAirportIdent is the Ident of the Approach procedure destination airport.

❑ FlightPlanApproachType (enum) [Get]

FlightPlanApproachType is an enum representing the type of approach procedure.

FlightPlanApproachType

#	Approach Type	#	Approach Type	#	Approach Type
0	UNKNOWN = 0	5	LORAN = 5	10	LDA = 10
1	VFR = 1	6	RNAV = 6	11	LOC = 11
2	HEL = 2	7	VOR = 7	12	MLS = 12
3	TACAN = 3	8	GPS = 8	13	ILS = 13
4	NDB = 4	9	SDF = 9		

http://msdn.microsoft.com/en-us/library/cc526954.aspx#AirportApproachType

In the KICT example, it is an ILS approach, FlightPlanApproachType = 13.

❑ FlightPlanApproachIndex (enum) [Get]

FlightPlanApproachIndex is an enum representing the selected approach for the destination airport. In the KICT example used throughout this section, ILS 19R approach has been selected, consequently FlightPlanApproachIndex = 3.

Approaches available for Wichita Mid-Continent Airport (KICT):

0 BLOC 19L
1 ILS 01L
2 ILS 01R
3 ILS 19R
4 NDB 01R (gps)
5 GPS 19L
6 GPS 32
7 RNAV 01L (gps)
8 RNAV 19R (gps)
9 VOR 14 (gps)

❑ FlightPlanApproachName (string) [Get]

FlightPlanApproachName is the name of the selected approach. In the KICT example, "**ILS 19R**".

❑ FlightPlanApproachTransitionIndex (enum) [Get]

FlightPlanApproachTransitionIndex is an enum representing the desired transition for the previously selected approach. In the KICT example used throughout this section, ILS 19R approach has been selected utilizing the ICT transition, consequently FlightPlanApproachTransitionIndex = 1.

Transitions available for the ILS 19R approach at Wichita Mid-Continent Airport (KICT):

0 VECTORS
1 ICT
2 HOVER

❑ FlightPlanApproachTransitionName (string) [Get]

FlightPlanApproachTransitionName is the name of the selected transition. In the KICT example, "**ICT**".

❑ FlightPlanIsApproachFinal (bool) [Get]

FlightPlanIsApproachFinal is a bool equaling 1 when FlightPlanWaypointApproachIndex 0 is the Final Approach Fix and WaypointApproachIndex 1 is the destination runway waypoint. The normal circumstance for this is a Vectors-To-Final Transition. Other than this situation, FlightPlanIsApproachFinal equals 0.

For a Vectors-To-Final Transition, fs9gps adds a 5.00 NMile sub-segment onto the Final Approach segment. The heading is the same as the Final Approach segment, and it is placed outboard of the Final Approach Fix. The sub-segment provides room to accommodate a turn to the Final Approach heading prior to reaching the Final Approach Fix.

Unlike the other sub-segments discussed in this section, the length of a Vectors-To-Final sub-segment is not a function of `flaps_up_stall_speed` specified in the aircraft.cfg file. It appears to always be 5.00 NMiles.

FlightPlanIsApproachFinal is set through use of FlightPlanNewApproachTransition or FlightPlanLoadApproach.

❑ FlightPlanIsApproachMissed (bool) [Get]

FlightPlanIsApproachMissed is a Boolean representing whether or not the Missed Approach procedure is included in the Approach procedure.

- ❑ 0 No Missed Approach procedure included
- ❑ 1 Missed Approach procedure is included in Approach

It is set through use of FlightPlanNewApproachMissed.

❑ FlightPlanApproachWaypointsNumber (enum) [Get]

FlightPlanApproachWaypointsNumber is the number of segments in the selected approach and transition. In the KICT ILS 19R / ICT Transition example used throughout this section, there are 9 approach segments (Index 0 through 8) going from the En route Fix that defines the Transition (Index 0) through the Holding Pattern at the Missed Approach Holding Waypoint (Index 8).

Note that when the sub-segments that define intra-segment turns are included, there is a combined total of 16 segments plus sub-segments.

❑ FlightPlanWaypointApproachIndex (enum) [Get, Set]

FlightPlanWaypointApproachIndex is the index pointer for the WaypointApproach variables. The WaypointApproach variables describe length, location, direction, etc., attributes of the approach segments.

❑ FlightPlanWaypointApproachICAO (string) [Get]

The ICAO ident of the currently indexed segment. Approach segments are defined by their termination points, so FlightPlanWaypointApproachICAO, ApproachName, ApproachLatitude and Longitude, and ApproachAltitude refer to the termination point of the segment.

❑ FlightPlanWaypointApproachName (string) [Get]

FlightPlanWaypointApproachName is the navaid ident or runway name associated with the termination point of the currently indexed approach segment.

❑ FlightPlanWaypointApproachType (enum) [Get]

FlightPlanWaypointApproachType is an enum referring to the navigation procedure objective of the currently indexed (but not necessarily the currently active) approach segment.

Waypoint Approach Type

Bit	Name and Type #	Bit	Name and Type #	Bit	Name and Type #
0	NONE = 0	4	DME_ARC_LEFT = 4	8	DISTANCE = 8
1	FIX = 1	5	DME_ARC_RIGHT = 5	9	ALTITUDE = 9
2	PROC_TURN_LEFT = 2	6	HOLDING_LEFT = 6	10	MANUAL_SEQ = 10
3	PROC_TURN_RIGHT = 3	7	HOLDING_RIGHT = 7	11	VECTORS_TO_FINAL = 11

http://msdn.microsoft.com/en-us/library/cc526954.aspx#GPS_APPROACH_WAYPOINT_TYPE

❑ FlightPlanWaypointApproachMode (enum) [Get]

FlightPlanWaypointApproachMode is a number used to describe the *currently indexed* segment of the approach.

Waypoint Approach Mode

Bit	Name and Type #	Bit	Name and Type #
0	NONE = 0	2	FINAL = 2
1	TRANSITION = 1	3	MISSED = 3

http://msdn.microsoft.com/en-us/library/cc526954.aspx#GPS_APPR_TYPE

- **FlightPlanWaypointApproachLatitude**
- **FlightPlanWaypointApproachLongitude (degrees) [Get]**

The latitude and longitude of the termination point – the waypoint - of the currently indexed approach segment. The termination point of the segment and the waypoint are one and the same.

- **FlightPlanWaypointApproachAltitude (feet) [Get]**

The altitude of the currently indexed approach waypoint.

- **FlightPlanWaypointApproachCourse (degrees) [Get]**

For straight approach segments, FlightPlanWaypointApproachCourse is the bearing of the approach segment pointing toward the termination point.

Unfortunately, fs9gps returns a mixture of true and magnetic bearings in what appears to be a software bug. They should be magnetic. Interestingly, the CustomDraw utility within fs9gps renders the flight path correctly, but the ApproachCourse variable that an autopilot accesses is not always the magnetic course that an autopilot expects, causing some turns to initially be over or under executed.

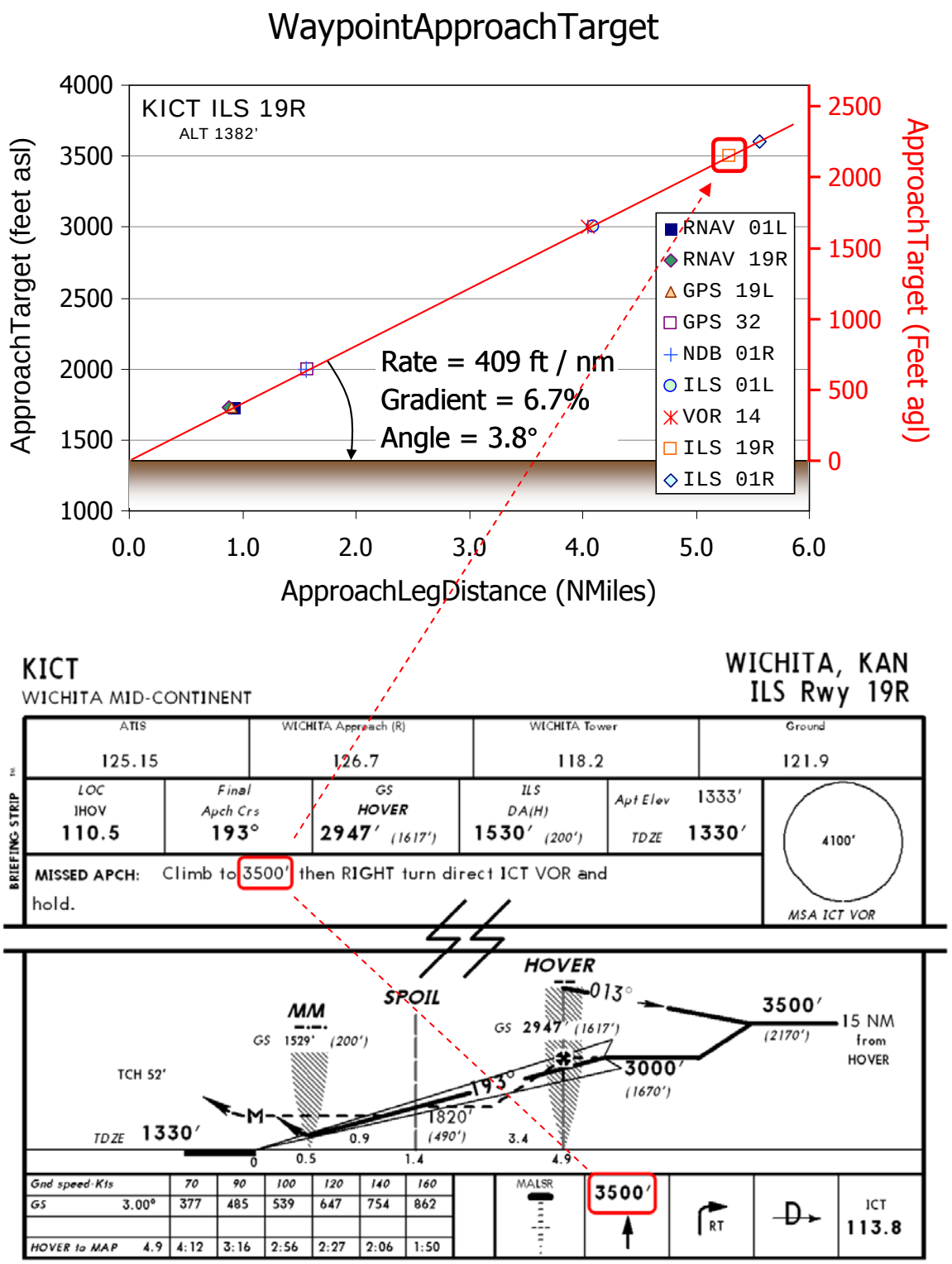
For segments that include turn sub-segments, the following applies:

- **Procedure Turn**: ApproachCourse is the magnetic bearing of the 45° turn. In the KICT ILS Rwy 19R example, 328°. Fs9gps correctly returns 328° M.
- **Missed Approach Turn toward Holding Fix**: ApproachCourse is the bearing of the culmination of the Missed Approach turn. Fs9gps returns a true bearing for this sub-segment. In the KICT ILS 19R Missed Approach example, this causes the aircraft to turn 7.1° too far to the north before ultimately turning direct to the Holding Fix, resulting in a "U" shaped Missed Approach Turn being flown rather than the smooth turn intended.
- **Holding Pattern Turn**: ApproachCourse is the bearing of the final sub-segment of the Holding Pattern, the segment that terminates at the Holding Fix. In the KICT ILS Rwy 19R example, it is 180° M. Fs9gps correctly returns 180°M.

The WaypointApproachCourse bearings listed in *green italic* font in the FS9 Transitions and Approach Segments ILS 19R approach diagram (Page 158) are degrees True, but fs9gps should have returned degrees Magnetic.

- **FlightPlanWaypointApproachTarget (feet) [Get]**

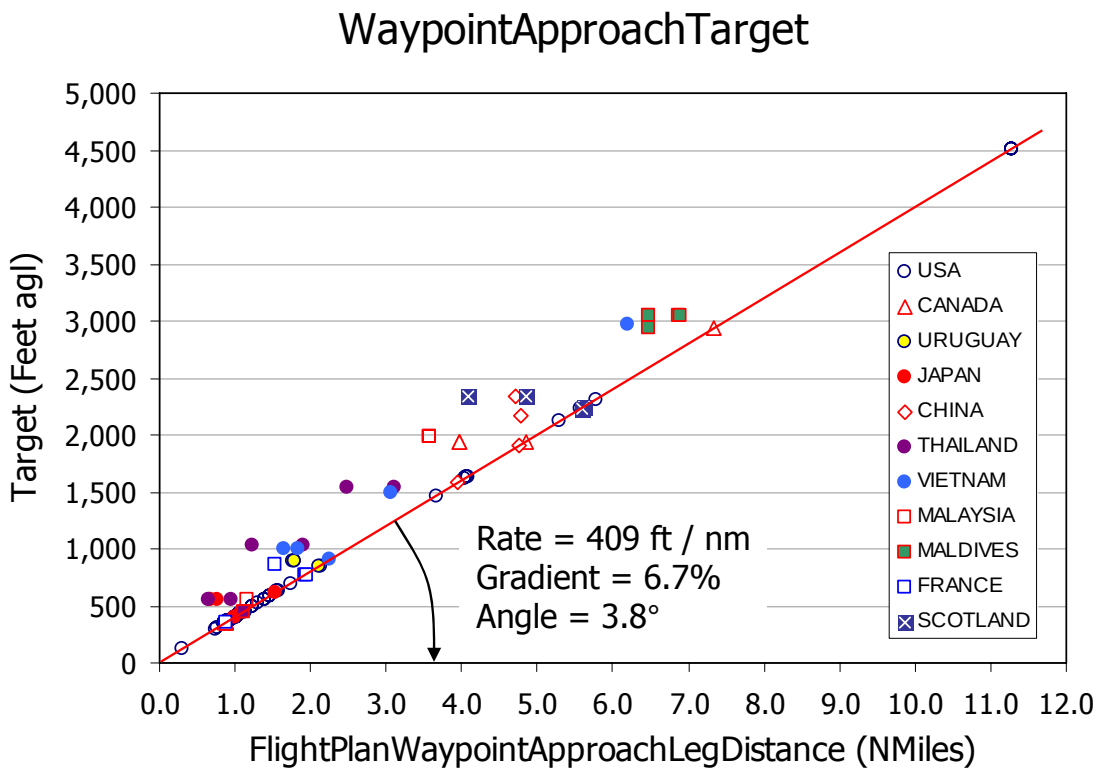
FlightPlanWaypointApproachTarget is the Missed Approach straight climb out target altitude. It is a function of ApproachLegDistance, at about a 3.8° climb angle, as shown in the graph below. The graph plots the ApproachTarget - LegDistance pairs for the Wichita, Kansas U.S.A. (KICT) airport approaches.



The approach segment that contains WaypointApproachTarget will be the first approach segment with WaypointApproachMode = 3. It may or may not have an ICAO or Name. At minimum, the Target waypoint has Lat and Lon, Type and Mode, which are sufficient information to define an Approach Waypoint.

In practical fs9gps terms, WaypointApproachTarget may be the altitude that FS9 ATC gives in its missed approach instructions.

The figure below shows a very small sample of WaypointApproachTarget from various approaches around the world. Many have a steeper Missed Approach climb-out rate, but it appears that there is a consistent minimum rate of 409 ft / NMile = 3.8°. There is also a noteworthy population of low altitude WaypointApproachTarget values. As a rule, an aircraft should not turn on climb-out under 400 feet agl, but that doesn't really explain why fs9gps has so many WaypointApproachTarget values around 500 feet agl.



❑ FlightPlanWaypointApproachLegDistance (nmiles) [Get]

FlightPlanWaypointApproachLegDistance is the length of the currently indexed approach segment. For approach segments that involve turns within the segment, such as a procedure turn, ApproachLegDistance is the combined flight distance of all parts, or sub-segments, of the approach segment, including the turn. Refer to the KICT ILS Rwy 19R ICT Transition segment discussion at the end of this section for further clarification.

FlightPlanWaypointApproachIndex = 0 is the En Route Fix and represents the starting point of the approach transition. ApproachLegDistance for this point is 0.00 before the approach is activated. Once activated, ApproachLegDistance for WaypointApproachIndex = 0 becomes the remaining distance from the aircraft position to the En Route Fix. This is the same as FlightPlanWaypointRemainingDistance of the active waypoint when the approach is activated.

❑ FlightPlanWaypointApproachLegTotalDistance (nmiles) [Get]

FlightPlanWaypointApproachLegTotalDistance is the cumulative distance of all Approach segments starting at WaypointApproachIndex = 0 through the currently indexed Approach Waypoint. When the index points to the last Approach Waypoint (the Holding fix usually), LegTotalDistance is the total length of the flight measured along flight segments, including one circuit around the holding pattern.

FlightPlanWaypointApproachLegTotalDistance of the Approach phase is analogous to FlightPlanWaypointDistanceTotal of the en route phase.

❑ FlightPlanWaypointApproachLegFromDistance (nmiles) [Get]

FlightPlanWaypointApproachLegFromDistance is the distance from the currently indexed Approach Segment to the termination point of the last Approach Segment.

FlightPlanWaypointApproachLegFromDistance of the Approach phase is analogous to FlightPlanWaypointDistanceRemaining of the en route phase.

❑ FlightPlanWaypointApproachRemainingDistance (nmiles) [Get]

FlightPlanWaypointApproachRemainingDistance is the segment distance remaining to be flown. For the active approach segment, that is, for the segment currently being flown, it is the remaining distance from the aircraft's current position to the termination point of the current segment. For segments beyond that, it is just the total length of the approach segment. For approach segments already passed, WaypointApproachRemainingDistance is 0.0.
FlightPlanWaypointApproachRemainingDistance is a segment measurement; it does not measure distances of *sub*-segments. FlightPlanApproachSegmentDistance does that.

FlightPlanWaypointApproachRemainingDistance of the Approach phase is analogous to FlightPlanWaypointWaypointRemainingDistance of the en route phase.

❑ FlightPlanWaypointApproachRemainingTotalDistance (nmiles) [Get]

FlightPlanWaypointApproachRemainingTotalDistance is the cumulative remaining distance from current aircraft position to the termination point of the indexed segment. It is the same as ApproachRemainingDistance when the indexed segment is the active segment. ApproachRemainingDistance is measured along flight plan segments; it is not a direct-to measurement.

FlightPlanWaypointApproachRemainingTotalDistance of the Approach phase is analogous to FlightPlanWaypointRemainingTotalDistance of the en route phase.

Miscellaneous

DISSECTING THE KICT ILS19R APPROACH SEGMENTS

A closer look at the construction of the approach segments found in the KICT example.

The table below lists the 9 waypoints associated with the ILS 19R Approach, ICT transition into KICT. Variable Segment and Sub-segment lengths are based on Flaps Up Stall Speed = 86.0 knots.

FLIGHT PLAN NEW APPROACH: KICT

```
      9 :ApprWaypointsNumber             13 :FlightPlanApprType          8 :FlightPlanWaypointApproachIndex
 ILS 19R :FlightPlanApprName            ICT :FlightPlanApprTransName     0 :FlightPlanActiveApproachWaypoint
      0 :FlightPlanIsActiveApproach       3 :FlightPlanApproachIndex     1 :FlightPlanApproachTransitionIndex
```

FlightPlanWaypointApproach

Idx	ICAO 111 123456789012	Name	Type	Mode	Latitude (Deg Min)	Longitude (Deg Min)	Latitude (Degrees)	Longitude (Degrees)	Alt	Trgt	Leg Dist	Course (mag)
0	VK3 ICT	ICT	1	1	37 44.7140	-97 35.0295	37.745233	-97.583825	0	0	0.00	-1.00
1	WK3KICTHOVER	HOVER	1	1	37 44.2538	-97 23.9393	37.737564	-97.398989	3500	0	8.78	93.06
2	WK3KICTHOVER	HOVER	3	1	37 58.6505	-97 17.8293	37.977508	-97.297155	3500	0	21.35	328.00
3	WK3KICTCF19R	CF19R	1	2	37 50.1523	-97 21.2320	37.835872	-97.353867	3500	0	8.91	197.52
4	WK3KICTHOVER	HOVER	1	2	37 44.2538	-97 23.9393	37.737564	-97.398989	3000	0	6.28	193.00
5	RK3KICTRW19R	RW19R	1	2	37 39.6962	-97 26.0290	37.661604	-97.433817	1382	0	4.85	193.00
6			9	3	37 34.6886	-97 28.2045	37.578144	-97.470076	3500	3500	5.30	193.00
7	VK3 ICT	ICT	1	3	37 44.7140	-97 35.0295	37.745233	-97.583825	3500	0	13.81	344.34
8	VK3 ICT	ICT	6	3	37 44.7140	-97 35.0295	37.745233	-97.583825	3500	0	14.34	180.00

The waypoint segments, sub-segments and flight path are demonstrated as follows:

FlightPlanWaypointApproachIndex 0:

Enroute Fix

FlightPlanActiveApproachWaypoint = 0 (the Index)
FlightPlanWaypointApproachType = 1 (Fix)
FlightPlanWaypointApproachMode = 1 (Transition)
FlightPlanApproachIsWaypointRunway = 0

En Route Fix Segment						0 Index	
	FlightPlanWaypointApproach					Leg	
Segment	Type	Mode	Altitude	Target	Course	Distance	
Enroute Fix	1	1	N/A (0')	N/A (0')	-001°	0.00	Enroute Fix (often is the Transition name)

Index 0 is the En Route Fix, which for this Approach Transition is the ICT VOR-DME.

FlightPlanWaypointApproachIndex 1:

Approach Transition

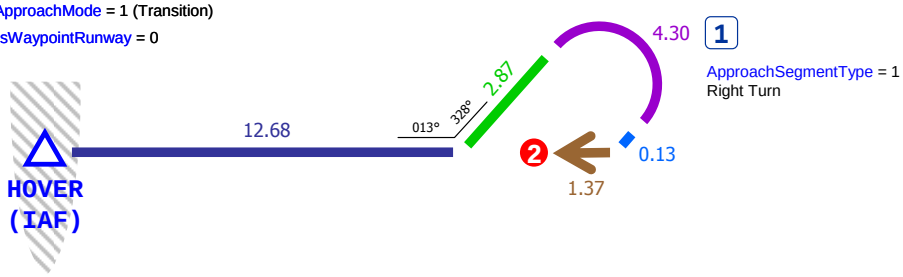
FlightPlanActiveApproachWaypoint = 1 (the Index)
FlightPlanWaypointApproachType = 1 (Fix)
FlightPlanWaypointApproachMode = 1 (Transition)
FlightPlanApproachIsWaypointRunway = 0

Approach Transition Segment						1 Index	
	FlightPlanWaypointApproach					Leg	
Segment	**Type**	**Mode**	**Altitude**	**Target**	**Course**	**Distance**	
Appr. Transition	1	1	3500	N/A (0')	093°	8.78	Begins at Enroute Fix. Ends at IAF - HOVER
						8.78 nm	:FlightPlanWaypointApproachLegDistance

FlightPlanWaypointApproachIndex 2:

Initial Approach

FlightPlanActiveApproachWaypoint = 2 (the Index)
FlightPlanWaypointApproachType = 3 (Procedure Turn Right)
FlightPlanWaypointApproachMode = 1 (Transition)
FlightPlanApproachIsWaypointRunway = 0

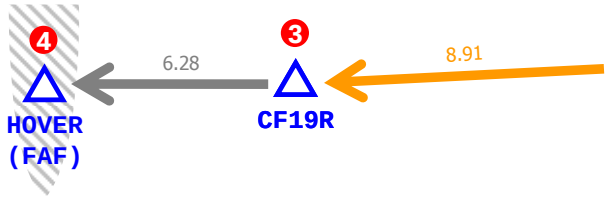


Initial Approach Segment						2 Index	
	FlightPlanWaypointApproach					Leg	
Sub-Segment	**Type**	**Mode**	**Altitude**	**Target**	**Course**	**Distance**	
Outbound	3	1	3500 ft	N/A (0')		12.68	Begins at Initial Approach Fix (IAF) - HOVER
45° Turn	3	1	3500 ft	N/A (0')	328°	2.87	328° = FlightPlanWaypointApproachCourse
180° Turn	3	1	3500 ft	N/A (0')		4.30	Right Turn. Note: ApproachSegmentType =1
45° Intercept	3	1	3500 ft	N/A (0')		0.13	
Inbound	3	1	3500 ft	N/A (0')		1.37	Ends at Intermed. Fix (IF) or Inbound to FAF
						21.35 nm	:FlightPlanWaypointApproachLegDistance

FlightPlanWaypointApproach Index 3 and 4:

Intermediate Approach

FlightPlanActiveApproachWaypoint = 3 and 4 (the Index)
FlightPlanWaypointApproachType = 1 (Fix)
FlightPlanWaypointApproachMode = 2 (Final)
FlightPlanApproachIsWaypointRunway = 0



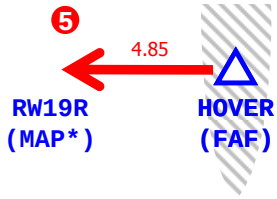
Intermediate Approach Segment 1						3 Index	
	FlightPlanWaypointApproach					Leg	
Segment	Type	Mode	Altitude	Target	Course	Distance	
Intermediate - 1	1	2	3500 ft	N/A (0')	198°	8.91	Begins at Intermed. Fix (IF) or Inbound to FAF
						8.91 nm	:FlightPlanWaypointApproachLegDistance

Intermediate Approach Segment 2						4 Index	
	FlightPlanWaypointApproach					Leg	
Segment	Type	Mode	Altitude	Target	Course	Distance	
Intermediate - 2	1	2	3000 ft	N/A (0')	193°	6.28	Ends at Final Approach Fix (FAF)
						6.28 nm	:FlightPlanWaypointApproachLegDistance

FlightPlanWaypointApproach Index 5:

Final Approach

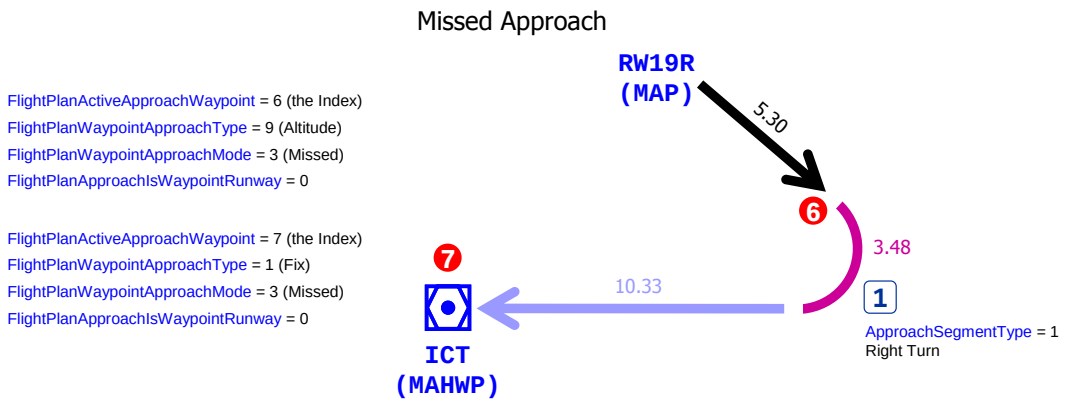
FlightPlanActiveApproachWaypoint = 5 (the Index)
FlightPlanWaypointApproachType = 1 (Fix)
FlightPlanWaypointApproachMode = 2 (Final)
FlightPlanApproachIsWaypointRunway = 1



* The Missed Approach Point

Final Approach Segment						5 Index	
	FlightPlanWaypointApproach					Leg	
Segment	Type	Mode	Altitude	Target	Course	Distance	
Final	1	2	1382 ft	N/A (0')	193°	4.85	Final Approach. Ends at MAP or Landing
						4.85 nm	:FlightPlanWaypointApproachLegDistance

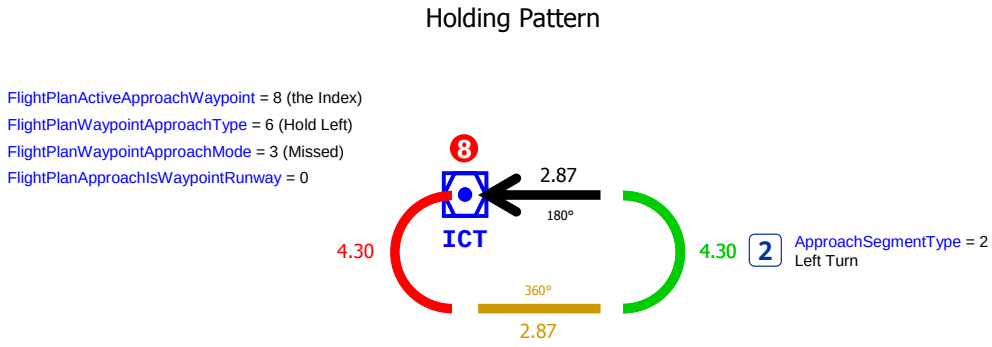
FlightPlanWaypointApproachIndex 6 and 7:



Missed Approach Climb-Out Segment						6 Index	
	FlightPlanWaypointApproach					Leg	
Segment	**Type**	**Mode**	**Altitude**	**Target**	**Course**	**Distance**	
Straight Climb-out	9	3	3500 ft	3500 ft	193°	5.30	Straight climb to Target altitude
						5.30 nm	:FlightPlanWaypointApproachLegDistance

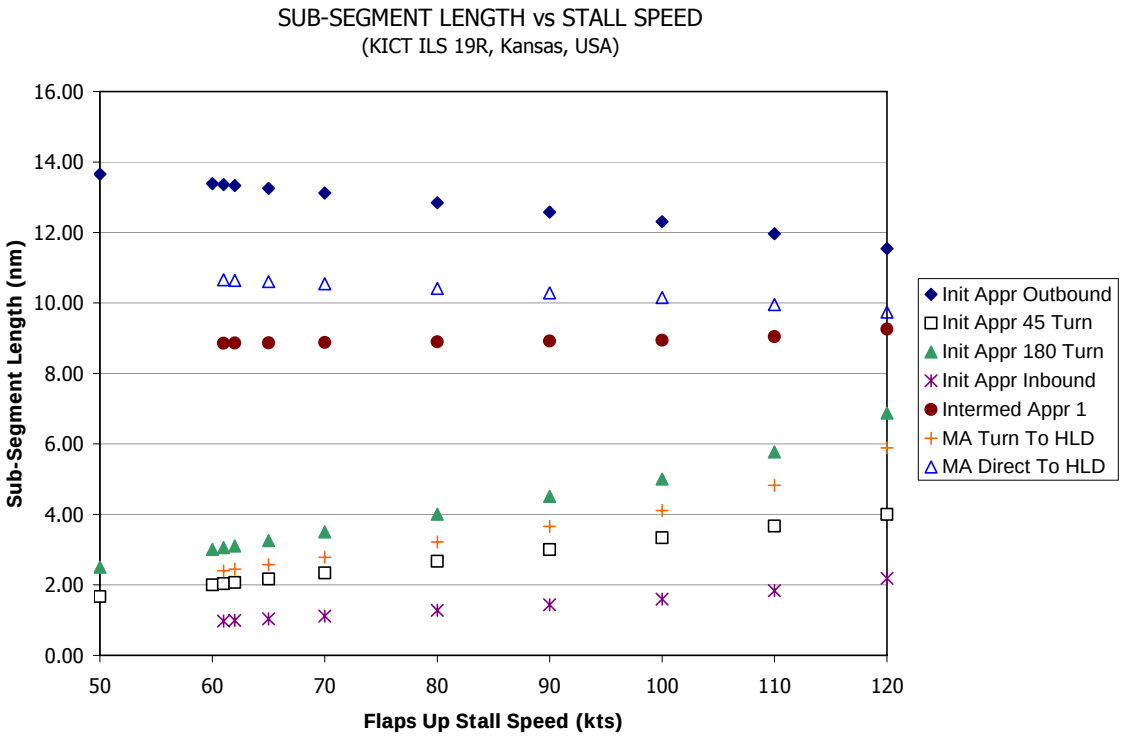
Missed Approach Holding Turn Segment						7 Index	
	FlightPlanWaypointApproach					Leg	
Sub-Segment	**Type**	**Mode**	**Altitude**	**Target**	**Course**	**Distance**	
Turn to Holding Fix	1	3	3500 ft	3500 ft	344°	3.48	Right turn toward Holding Fix. ApproachSegmentType =1
Direct to Holding Fix	1	3	3500 ft	3500 ft		10.33	Direct to Holding Fix.
						13.81 nm	:FlightPlanWaypointApproachLegDistance

FlightPlanWaypointApproachIndex 8:



Holding Pattern Segment						8 Index	
	FlightPlanWaypointApproach					Leg	
Sub-Segment	**Type**	**Mode**	**Altitude**	**Target**	**Course**	**Distance**	
180° Turn North	6	3	3500 ft	N/A (0')		4.30	Begins at MAHWP. ApproachSegmentType = 2
North leg	6	3	3500 ft	N/A (0')		2.87	
180° Turn South	6	3	3500 ft	N/A (0')		4.30	ApproachSegmentType = 2
South Leg	6	3	3500 ft	N/A (0')	180°	2.87	Ends at Missed Appr. Holding Waypoint (MAHWP)
						14.34 nm	:FlightPlanWaypointApproachLegDistance

SUB-SEGMENT LENGTH



The length of some approach sub-segments varies according to the flaps_up_stall_speed specified in the aircraft.cfg file. fs9gps apparently does this to accommodate the greater turning radius required as approach speeds increase.

The example above shows various sub-segment lengths associated with the KICT ILS19R Approach. The sub-segment names follow those used in the examples found in "**DISSECTING THE KICT ILS19R APPROACH SEGMENTS**".

Note that the turning sub-segment (e.g., ▲ Init Appr 180 Turn and + MA Turn To HLD) lengths increase with stall speed, while some of the straight sub-segments (e.g., ◆ Init Appr Outbound and △ MA Direct To HLD) decrease. The decrease is necessary to keep the overall procedure roughly the same size regardless of stall speed, and, in the case of a Procedure Turn, to try to keep the aircraft within the Manuevering Area. In the case of at least KICT ILS19R used as an example in this chapter, fs9gps does not comply with the requirement to complete the Procedure Turn within 15 NMiles of the IAF. The Initial Approach Outbound Leg sub-segment is too long.

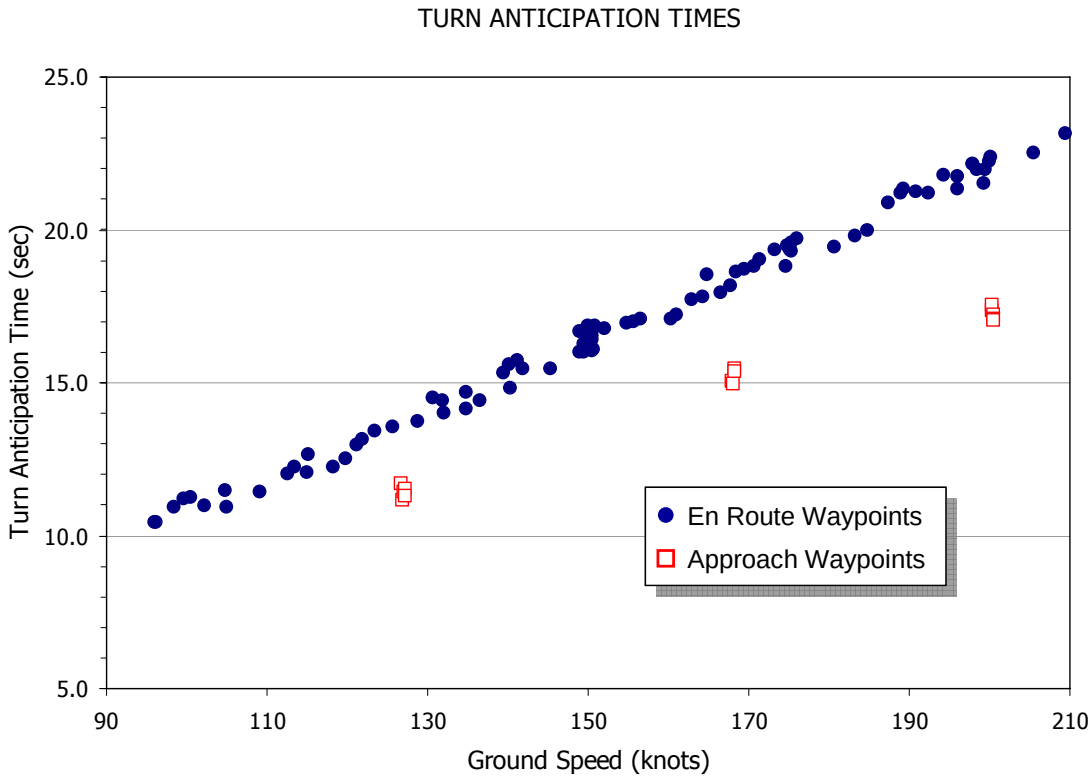
Other segments and some sub-segments are geographically fixed and do not vary by aircraft stall speed. KICT ILS19R examples include the Approach Transition, Intermediate Approach "2", Final Approach, and Missed Approach Straight Climb-out segments.

FLY-BY vs. FLY-OVER WAYPOINTS

Does fs9gps distinguish between Fly-By and Fly-Over Waypoints? Technically, no. If you stretch it … perhaps, but I think only to the extent that it distinguishes between En Route and Approach Waypoints.

Fly-Over Waypoints are usually associated with RNAV procedures, SIDs and STARs. I haven't studied enough fs9gps RNAV approaches to say if fs9gps treats RNAV Fly-over waypoints differently than RNAV Fly-By waypoints, but I suspect not.

However, I can say this much: the Turn Anticipation Time of Approach Waypoints, which include the typical Fly-Over Waypoints Missed Approach Point and Holding Pattern Fix, is less than that of En Route Waypoints, as shown in the chart below.

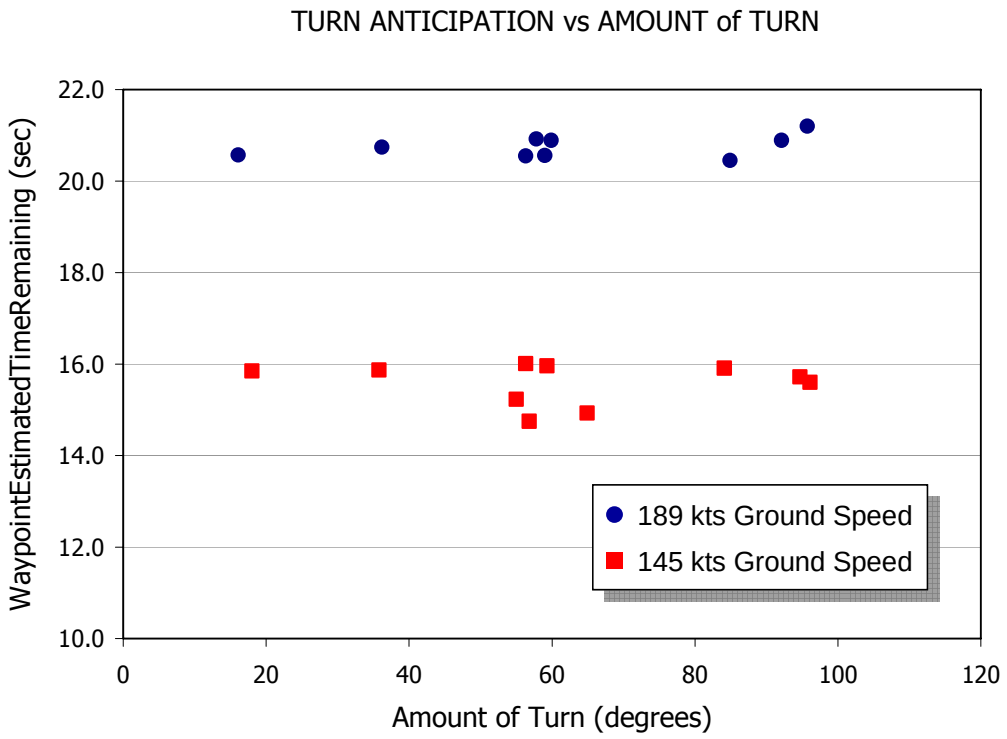


The smaller times result in more precise turns in the Approach phase when the aircraft should be traveling at approach rather than cruise airspeeds. So, closer tolerance on the Fly-Over Waypoints is part of the package.

TURN ANTICIPATION vs. AMOUNT OF TURN

Does the amount of turn (number of degrees turned) influence Turn Anticipation? No.

The chart below plots Turn Anticipation Time against Amount of Turn at two different Ground Speeds. The amount of turn has no impact on Turn Anticipation Time.



It's straight-forward for fs9gps to calculate remaining time to the next waypoint, but it might be a little more involved to adjust Turn Anticipation Time for the amount of turn ahead.

Most real-world Turn Anticipation Distance algorithms and rules of thumb are based on amount of turn, true airspeed, and bank angle. FS9's world is simpler.

FlightPlanWaypointFrequency DATAPOINTS

A few data points. A loose trend, un-predictable and non-sensical. As MSFT already stated, this variable is not implemented (not correctly anyway).