AIRCRAFT RADIO NAVIGATION VARIABLES

The tables below indicate the properties for the Simulation Variables that can be used to get and set properties related to the various radio navigation systems of an aircraft and Air Traffic Control. For information on the units listed for each variable, please see here: Simulation Variable Units

> **NOTE**: *When Microsoft Flight Simulator is running in multiplayer mode, only a small number of variables are communicated between aircraft. Those variables that are available will say so in the description as being either for "All Aircraft" or for "Shared Cockpit".*

You can find a complete index of all available SimVars here: SimVar Index

ADF

Simulation Variable	Description	Units	Settable
`ADF ACTIVE FREQUENCY:index`	*ADF* frequency. Index of 1 or 2.	Frequency ADF BCD32	
`ADF AVAILABLE:index`	True if ADF is available	Bool	
`ADF CARD`	ADF compass rose setting	Degrees	
`ADF EXT FREQUENCY` *Deprecated*	Deprecated, use `ADF ACTIVE FREQUENCY`	Frequency BCD16	

`ADF FREQUENCY` *Deprecated*	Deprecated, use `ADF ACTIVE FREQUENCY`	Frequency BCD16	
`ADF IDENT`	*ICAO code*	String	
`ADF LATLONALT:index`	Returns the latitude, longitude and altitude of the station the radio equipment is currently tuned to, or zeros if the radio is not tuned to any ADF station. Index of 1 or 2 for ADF 1 and ADF 2.	`SIMCONNECT_DATA_LATLONALT` structure	
`ADF NAME:index`	Descriptive name	String	
`ADF RADIAL:index`	Current direction from *NDB* station	Degrees	
`ADF RADIAL MAG:index`	Returns the magnetic bearing to the currently tuned ADF transmitter.	Degrees	
`ADF SIGNAL:index`	Signal strength	Number	
`ADF SOUND:index`	*ADF* audio flag. Index of 0 or 1.	Bool	

ADF STANDBY AVAILABLE:index	True if *ADF* Standby is available	Bool	
ADF STANDBY FREQUENCY:index	*ADF* standby frequency	Hz	
ADF VOLUME	Returns the volume of the *ADF*	*Percent Over 100*	

ATC

Simulation Variable	Description	Units	Settable
ATC AIRLINE	The name of the Airline used by ATC, as a string with a maximum length of 50 characters.	String	
ATC AIRPORT IS TOWERED	If the airport is controlled, this boolean is true.	Bool	
ATC CLEARED IFR	Returns whether or not the user has filed an IFR flightplan that has been cleared by the sim ATC	Bool	
ATC CLEARED LANDING	Whether the ATC has cleared the plane for landing.	Bool	
ATC CLEARED TAKEOFF	Whether the ATC has cleared the plane for takeoff.	Bool	
ATC CLEARED TAXI	Whether the ATC has cleared the plane for taxi.	Bool	
ATC CURRENT WAYPOINT ALTITUDE	Returns the target altitude for the current ATC flightplan waypoint.	Bool	

ATC FLIGHT NUMBER	Flight Number used by ATC, as a string with a maximum number of 6 characters.	String	
ATC FLIGHTPLAN DIFF ALT	Altitude between the position of the aircraft and his closest waypoints in the flightplan.	Meters	
ATC FLIGHTPLAN DIFF DISTANCE	Returns the lateral distance the user's plane is from the ATC flight plan track.	Meters	
ATC FLIGHTPLAN DIFF HEADING	Heading between the position of the aircraft and his closest waypoints in the flightplan.	Degrees	
ATC HEAVY	Is this aircraft recognized by ATC as heavy.	Bool	
ATC ID	ID used by ATC, as a string with a maximum number of 10 characters.	String	
ATC IFR FP TO REQUEST	Returns true if the user has a valid IFR flight plan they can as for clearance for with ATC at the airport they are currently at.	Bool	
ATC MODEL	Model used by ATC, as a string with a maximum number of 10 characters.	String	
ATC ON PARKING SPOT	Is ATC aircraft on parking spot.	Bool	
ATC PREVIOUS WAYPOINT ALTITUDE	Returns the target altitude for the previous ATC flightplan waypoint.	Meters	
ATC RUNWAY AIRPORT NAME	The name of the airport of the runway assigned by the ATC. Returns "" if no runway is assigned.	String	
ATC RUNWAY DISTANCE	This float represents the distance between the player's	Meters	

	plane and the center of the runway selected by the ATC.		
`ATC RUNWAY END DISTANCE`	This is a float corresponding to the horizontal distance between the player's plane and the end of the runway selected by the ATC.	Meters	
`ATC RUNWAY HEADING DEGREES TRUE`	This float represents the true heading of the runway selected by the ATC.	Degrees	
`ATC RUNWAY LENGTH`	The length of the runway assigned by the ATC. Returns -1 if no runway is assigned.	Meters	
`ATC RUNWAY RELATIVE POSITION X`	This is a float corresponding to the player's main gear relative X (transverse) position on the runway selected by the ATC.	Meters	
`ATC RUNWAY RELATIVE POSITION Y`	This is a float corresponding to the player's main gear relative Y (height) position on the runway selected by the ATC.	Meters	
`ATC RUNWAY RELATIVE POSITION Z`	This is a float corresponding to the player's main gear relative Z (longitudinal) position on the runway selected by the ATC.	Meters	
`ATC RUNWAY SELECTED`	This is a boolean corresponding to whether or not the ATC has pre-selected a runway for the player's plane. If this is false, every other `ATC RUNWAY *` SimVar will return default values.	Bool	
`ATC RUNWAY START DISTANCE`	This is a float corresponding to the horizontal distance between the player's plane and the start of the runway selected by the ATC.	Meters	

Simulation Variable	Description	Units	Settable
ATC RUNWAY TDPOINT RELATIVE POSITION X	This float represents the player's main gear relative X (transverse) position according to the aiming point of the runway selected by the ATC.	Meters	
ATC RUNWAY TDPOINT RELATIVE POSITION Y	This float represents the player's main gear relative Y (height) position according to the aiming point of the runway selected by the ATC.	Meters	
ATC RUNWAY TDPOINT RELATIVE POSITION Z	This float represents the player's main relative Z (longitudinal) position according to the aiming point of the runway selected by the ATC.	Meters	
ATC RUNWAY WIDTH	The width of the runway assigned by the ATC. Returns -1 if no runway is assigned.	Meters	
ATC SUGGESTED MIN RWY LANDING	Suggested minimum runway length for landing. Used by ATC.	Feet	
ATC SUGGESTED MIN RWY TAKEOFF	Suggested minimum runway length for takeoff. Used by ATC.	Feet	
ATC TAXIPATH DISTANCE	Returns the lateral distance the user's plane is from the path of the currently issued ATC taxi instructions.	Meters	
ATC TYPE	Type used by ATC.	String (30)	

COM

Simulation Variable	Description	Units	Settable

COM1 STORED FREQUENCY COM2 STORED FREQUENCY COM3 STORED FREQUENCY	The stored COM 1/2/3 frequency value.	Frequency BCD16	
COM ACTIVE BEARING:index	Gives the bearing (in degrees) of the active COM station (airport) or a value less than 0 if the station does not belong to an airport. Index is 1, 2 or 3.	Degrees	
COM ACTIVE DISTANCE:index	Gives the distance (in meters) to the active COM station (airport) or a value less than -180° if the station does not belong to an airport. Index is 1, 2 or 3.	Meters	
COM ACTIVE FREQUENCY:index	Com frequency. Index is 1, 2 or 3.	Frequency BCD16	
COM ACTIVE FREQ IDENT:index	The identity of the station that is tuned on the indexed active COM radio. Index is 1, 2, or 3.	String	

`COM ACTIVE FREQ TYPE:index`	The type of COM frequency for the active indexed COM system. Index is 1, 2, or 3.	String: "ATIS" - Atis "UNI" - Unicom "CTAF" - CTAF "GND" - Ground "TWR" - Tower "CLR" - Clearance Delivery "APPR" - Approach "DEP" - Departure "FSS" - FSS "AWS" - AWOS	
`COM ACTIVE LATLONALT:index`	This will return the latitude, longitude and altitude corresponding to the indexed COM station associated with the active COM frequency. If the station is not associated with an airport, then the lat/lon/alt values returned will be `-15943°, 80°, -10000` (this means that you can simply check that the altitude value is greater than 0 to assure the validity of the returned struct). Index is 1, 2 or 3.	Struct: `SIMCONNECT_DATA_LATLONALT`	
`COM AVAILABLE:index`	True if COM1, COM2 or COM3 is available	Bool	

	(depending on the index, either 1, 2, or 3)		
`COM LATLONALT:index`	*Not currently used in the simulation.*	Struct: `SIMCONNECT_DATA_LATLONALT`	
`COM RECEIVE:index`	Whether or not the plane is receiving on the indexed com channel or not (either 1, 2, or 3 for the index).	Bool	
`COM RECEIVE ALL`	Toggles all COM radios to receive on	Bool	
`COM RECEIVE EX1:index`	Whether or not the plane is receiving on the indexed com channel. Index is 1, 2 or 3.	Bool	

`COM SPACING MODE:index`	The COM radio frequency step. Index is 1, 2 or 3.	Enum:
		
		   0 = 25kHz
		   1 = 8.33kHz	

`COM STANDBY FREQUENCY:index`	Com standby frequency. Index is 1, 2 or 3.	Frequency BCD16	
`COM STANDBY FREQ IDENT:index`	The identity of the station that is tuned on the indexed standby COM radio. Index is 1, 2, or 3.	String	

`COM STANDBY FREQ TYPE:index`	The type of COM frequency for the standby	String:
		
		   "ATIS" - Atis
		   "UNI" - Unicom	

Simulation Variable	Description	Units	Settable
	indexed COM system. Index is 1, 2, or 3.	"CTAF" - CTAF "GND" - Ground "TWR" - Tower "CLR" - Clearance Delivery "APPR" - Approach "DEP" - Departure "FSS" - FSS "AWS" - AWOS	
`COM STATUS:index`	Radio status flag for the indexed com channel. Index is 1, 2 or 3.	Enum: -1 = Invalid 0 = OK 1 = Does not exist 2 = No electricity 3 = Failed	
`COM TEST:index`	Enter an index of 1, 2 or 3. Will return `TRUE` if the COM system is working, `FALSE` otherwise.	Bool	
`COM TRANSMIT:index`	Audio panel com transmit state. Index of 1, 2 or 3.	Bool	
`COM VOLUME`	The volume of the COM Radio.	Percent	

FLARM

Simulation Variable	Description	Units	Settable
`FLARM AVAILABLE`	Whether the FLARM is available (TRUE, 1) or not (FALSE, 0).	Bool	

FLARM THREAT BEARING	The bearing of the FLARM threat aircraft, relative to track.	Degrees	
FLARM THREAT DATA	The FLARM threat aircraft data structure, which contains data about the perceived threat, returned as a struct. Struct member variables are as follows: `id` (U62): the network id of the intruding plane so that they are remembered in order to compute their trajectory. `bearing` (FLOAT64): The threat bearing, in degrees (this is bearing from track axis and not bearing from the airplane axis). `heading` (FLOAT64): The threat heading. `distance` (FLOAT64): The distance between the aircraft and the threat, in meters. `verticalBearing` (FLOAT64): The vertical bearing between the aircraft and the threat, in degrees. `relativeAltitude` (FLOAT64): The relative altitude of the threat to the aircraft, in meters. `timeToCollision` (FLOAT64): The estimated time to a collision, in seconds.	Struct	
FLARM THREAT DISTANCE	The distance to the FLARM threat object.	Meters	
FLARM THREAT HEADING	The heading to the FLARM threat object.	Degrees	
FLARM THREAT RELATIVE ALTITUDE	The relative altitude of the threat object.	Meters	
FLARM THREAT TIME TO COLLISION	The estimated time to a collision.	Seconds	
FLARM THREAT VERTICAL BEARING	The vertical bearing towards the threat.	Degrees	

GPS

When using these SimVars, it is worth noting that to correctly set those that can be set, you should first use the SimVar `GPS OVERRIDEN` and set it to `TRUE`. If you don't set this SimVar, then any changes that are made to the other GPS variables will be overriden by the simulation GPS system moments after you set them.

Simulation Variable	Description	Units	Settable
`GPS APPROACH AIRPORT ID`	ID of airport.	String	
`GPS APPROACH APPROACH ID`	ID of approach.	String	
`GPS APPROACH APPROACH INDEX`	Index of approach for given airport.	Number	
`GPS APPROACH APPROACH TYPE`	Approach type.	Enum: 0 = None 1 = GPS 2 = VOR 3 = NDB 4 = ILS 5 = Localizer 6 = SDF 7 = LDA 8 = VOR/DME 9 = NDB/DME 10 = RNAV 11 = Backcourse	
`GPS APPROACH IS FINAL`	Is approach transition final approach segment.	Bool	
`GPS APPROACH IS MISSED`	Is approach segment missed approach segment.	Bool	

GPS APPROACH IS WP RUNWAY	Waypoint is the runway.	Bool	
GPS APPROACH MODE	Sub mode within approach mode.	Enum: 0 = None 1 = Transition 2 = Final 3 = Missed	
GPS APPROACH SEGMENT TYPE	Segment type within approach.	Enum: 0 = Line 1 = Arc clockwise 2 = Arc counter-clockwise	
GPS APPROACH TIMEZONE DEVIATION	Deviation of local time from GMT.	Seconds	
GPS APPROACH TRANSITION ID	ID of approach transition.	String	
GPS APPROACH TRANSITION INDEX	Index of approach transition.	Number	
GPS APPROACH WP COUNT	Number of waypoints.	Number	
GPS APPROACH WP INDEX	Index of current waypoint.	Number	
GPS APPROACH WP TYPE	Waypoint type within approach mode.	Enum: 0 = None 1 = Fix 2 = Procedure turn left 3 = Procedure turn right 4 = Dme arc left 5 = Dme arc right 6 = Holding left	

		7 = Holding right	
		8 = Distance	
		9 = Altitude	
		10 = Manual sequence	
		11 = Vector to final	
GPS CDI NEEDLE	The course deviation of the needle for a CDI instrument. The SimVar displays the deviation from -127 to +127. It returns a value if a flight plan is set (otherwise it will return 0) even if the autopilot isn't on GPS mode. Scaling can also be set through the GPS CDI SCALING simvar.	Number	
GPS CDI SCALING	The full scale deflection of the CDI due to GPS cross-track error, in meters.	Meters	
GPS COURSE TO STEER	Suggested heading to steer (for autopilot).	Radians	
GPS DRIVES NAV1	GPS is driving Nav 1 indicator. Note this setting will also affect the SimVars HSI_STATION_IDENT and HSI_BEARING.	Bool	
GPS ETA	Estimated time of arrival at destination.	Seconds	
GPS ETE	Estimated time en route to destination.	Seconds	
GPS FLIGHTPLAN TOTAL DISTANCE	This is the complete flightplan length from start to end. Essentially the cumulative length of	Meters	

	all the flight plan legs added together.		
`GPS FLIGHT PLAN WP COUNT`	Number of waypoints.	Number	
`GPS FLIGHT PLAN WP INDEX`	Index of waypoint.	Number	
`GPS GROUND MAGNETIC TRACK`	Current magnetic ground track.	Radians	
`GPS GROUND SPEED`	Current ground speed.	Meters per second	
`GPS GROUND TRUE HEADING`	Current true heading.	Radians	
`GPS GROUND TRUE TRACK`	Current true ground track.	Radians	
`GPS GSI SCALING`	The full scale deflection of the vertical GSI due to GPS glidepath deviation, in meters.	Meters	
`GPS HAS GLIDEPATH`	Whether or not the GPS system has a presently available glidepath for guidance. Only applicable with `GPS_OVERRIDDEN`. When true and in `GPS OVERRIDDEN`, `HSI_GSI_NEEDLE_VALID` will also be true.	Bool	
`GPS HSI NEEDLE`	The glide deviation of the needle for a CDI instrument. The simvar displays the deviation from -127 to +127. It returns a value if a flight plan is set (otherwise it will return 0) even if the autopilot isn't on GPS mode. Scaling can also be	Number	

	set through the `GPS CDI SCALING` simvar.		
`GPS IS ACTIVE FLIGHT PLAN`	Flight plan mode active.	Bool	
`GPS IS ACTIVE WAY POINT`	Waypoint mode active.	Bool	
`GPS IS ACTIVE WP LOCKED`	Is switching to next waypoint locked.	Bool	
`GPS IS APPROACH ACTIVE`	Is approach mode active.	Bool	
`GPS IS APPROACH LOADED`	Is approach loaded.	Bool	
`GPS IS ARRIVED`	Is flight plan destination reached.	Bool	
`GPS IS DIRECTTO FLIGHTPLAN`	Is Direct To Waypoint mode active.	Bool	
`GPS MAGVAR`	Current GPS magnetic variation.	Radians	
`GPS OBS ACTIVE`	Whether or not the OBS mode is currently active (disable the automatic sequencing of waypoints in GPS flight plan).	Bool	
`GPS OBS VALUE`	This is the currently selected OBS course in degrees, from 0 to 360.	Degrees	
`GPS OVERRIDDEN`	When it is active, all sim GPS system updates are suspended. This must be set to TRUE to be able to correctly set to any other GPS SimVar.	Bool	

`GPS POSITION ALT`	Current GPS altitude.	Meters	
`GPS POSITION LAT`	Current GPS latitude.	Degrees	
`GPS POSITION LON`	Current GPS longitude.	Degrees	
`GPS TARGET ALTITUDE`	Altitude of GPS target.	Meters	
`GPS TARGET DISTANCE`	Distance to target.	Meters	
`GPS VERTICAL ANGLE`	Glidepath in degrees.	Degrees	
`GPS VERTICAL ANGLE ERROR`	Vertical error in degrees from GlidePath.	Degrees	
`GPS VERTICAL ERROR`	Vertical deviation in meters from GlidePath.	Meters	
`GPS WP BEARING`	Magnetic bearing to waypoint.	Radians	
`GPS WP CROSS TRK`	Cross track distance.	Meters	
`GPS WP DESIRED TRACK`	The required heading (magnetic) from the previous waypoint to the next waypoint.	Radians	
`GPS WP DISTANCE`	Distance to waypoint.	Meters	
`GPS WP ETA`	Estimated time of arrival at waypoint.	Seconds	
`GPS WP ETE`	Estimated time en route to waypoint.	Seconds	
`GPS WP NEXT ALT`	Altitude of next waypoint.	Meters	

`GPS WP NEXT ID`	ID of next GPS waypoint.	String	
`GPS WP NEXT LAT`	Latitude of next waypoint.	Degrees	
`GPS WP NEXT LON`	Longitude of next waypoint.	Degrees	
`GPS WP PREV ALT`	Altitude of previous waypoint.	Meters	
`GPS WP PREV ID`	ID of previous GPS waypoint.	String	
`GPS WP PREV LAT`	Latitude of previous waypoint.	Degrees	
`GPS WP PREV LON`	Longitude of previous waypoint.	Degrees	
`GPS WP PREV VALID`	Is previous waypoint valid (i.e. current waypoint is not the first waypoint).	Bool	
`GPS WP TRACK ANGLE ERROR`	Tracking angle error to waypoint.	Radians	
`GPS WP TRUE BEARING`	True bearing to waypoint.	Radians	
`GPS WP TRUE REQ HDG`	Required true heading to waypoint.	Radians	
`GPS WP VERTICAL SPEED`	Vertical speed to waypoint.	Meters per second	

HSI

Simulation Variable	Description	Units	Settable
`HSI BEARING`	If the `GPS_DRIVES_NAV1` variable is true and the `HSI`	Degrees	

	`BEARING VALID` variable is true, this variable contains the HSI needle bearing. If the `GPS DRIVES NAV1` variable is false and the `HSI BEARING VALID` variable is true, this variable contains the ADF1 frequency.		
`HSI BEARING VALID`	This will return true if the `HSI BEARING` variable contains valid data.	Bool	
`HSI CDI NEEDLE`	Needle deflection (+/- 127).	Number	
`HSI CDI NEEDLE VALID`	Signal valid.	Bool	
`HSI DISTANCE`	DME/GPS distance.	Nautical miles	
`HSI GSI NEEDLE`	Needle deflection (+/- 119).	Number	
`HSI GSI NEEDLE VALID`	Signal valid.	Bool	
`HSI HAS LOCALIZER`	Station is a localizer.	Bool	
`HSI SPEED`	DME/GPS speed.	Knots	
`HSI STATION IDENT`	Returns the ident of the the next GPS waypoint, if `GPS_DRIVES_NAV1` is true. If `GPS DRIVES NAV1` is false, it returns the identity of the station that is tuned on nav radio 1.	String	
`HSI TF FLAGS`	Nav TO/FROM flag.	Enum: 0 = Off 1 = TO 2 = FROM	

MARKERS

Simulation Variable	Description	Units	Settable
INNER MARKER	Inner marker state.	Bool	
INNER MARKER LATLONALT	Returns the latitude, longitude and altitude of the inner marker of an approach to a runway, if the aircraft is within the required proximity, otherwise it will return zeros.	SIMCONNECT_DATA_LATLONALT structure	
MARKER AVAILABLE	True if Marker is available.	Bool	
MARKER BEACON SENSITIVITY HIGH	Whether or not the Marker Beacon is in High Sensitivity mode.	Bool	
MARKER BEACON STATE	Marker beacon state.	Enum: 0 = None 1 = Outer 2 = Middle 3 = Inner	
MARKER BEACON TEST MUTE	Whether or not the Marker Beacon is in Test/Mute mode.	Bool	
MARKER SOUND	Marker audio flag.	Bool	

MIDDLE MARKER	Middle marker state.	Bool	
MIDDLE MARKER LATLONALT	Returns the latitude, longitude and altitude of the middle marker.	SIMCONNECT_DATA_LATLONALT structure	
OUTER MARKER	Outer marker state.	Bool	
OUTER MARKER LATLONALT	Returns the latitude, longitude and altitude of the outer marker.	SIMCONNECT_DATA_LATLONALT structure	

NAV

Simulation Variable	Description	Units	Settable
NAV ACTIVE FREQUENCY:index	Nav active frequency. Index is 1 or 2.	MHz	
NAV AVAILABLE:index	Flag if Nav equipped on aircraft.	Bool	
NAV BACK COURSE FLAGS:index	Returns the listed bit flags.	Flags: BIT0:[index] 1=back course available BIT1:[index] 1=localizer tuned in BIT2:[index] 1=on course BIT7:[index] 1=station active	
NAV CDI:index	CDI needle deflection (+/- 127).	Number	
NAV CLOSE DME:index	Closest DME distance. Requires an index value from 1 to 4 to set which NAV to target.	Nautical miles	

	Note that this SimVar will only work if the `NAV1_CLOSE_FREQ_SET` key event has been set to 1 (TRUE).		
`NAV CLOSE FREQUENCY:index`	Closest Localizer course frequency. Requires an index value from 1 to 4 to set which NAV to target. Note that this SimVar will only work if the `NAV1_CLOSE_FREQ_SET` key event has been set to 1 (TRUE).	Hz	
`NAV CLOSE IDENT:index`	ICAO code. Requires an index value from 1 to 4 to set which NAV to target. Note that this SimVar will only work if the `NAV1_CLOSE_FREQ_SET` key event has been set to 1 (TRUE).	String	
`NAV CLOSE LOCALIZER:index`	Closest Localizer course heading. Requires an index value from 1 to 4 to set which NAV to target. Note that this SimVar will only work if the `NAV1_CLOSE_FREQ_SET` key event has been set to 1 (TRUE).	Degrees	
`NAV CLOSE NAME:index`	Descriptive name. Requires an index value from 1 to 4 to set which NAV to target. Note that this SimVar will only work if the	String	

	`NAV1_CLOSE_FREQ_SET` key event has been set to 1 (TRUE).		
`NAV CODES`	Returns bit flags with the listed meaning.	Flags: BIT7:[index] 0= VOR 1= Localizer BIT6:[index] 1= glideslope available BIT5:[index] 1= no localizer backcourse BIT4:[index] 1= DME transmitter at glide slope transmitter BIT3:[index] 1= no nav signal available BIT2:[index] 1= voice available BIT1:[index] 1 = TACAN available BIT0:[index] 1= DME available	
`NAV DME`	DME distance.	Nautical miles	
`NAV DMESPEED`	DME speed.	Knots	
`NAV DME LATLONALT:index`	Returns the DME station.	`SIMCONNECT_DATA_LATLONALT` structure	
`NAV FREQUENCY`	Localizer course frequency	Hz	
`NAV GLIDE SLOPE`	The glide slope gradient. The value returned is an integer value formed as follows: `sin(slope) * 65536 * 2` So, for example, a glide slope of 2.7º would return a value of 6174. TO get the value in degrees, then	Number	

	use `NAV_RAW_GLIDE_SLOPE` instead.		
`NAV GLIDE SLOPE ERROR`	Difference between current position and glideslope angle. Note that this provides 32 bit floating point precision, rather than the 8 bit integer precision of NAV GSI.	Degrees	
`NAV GLIDE SLOPE LENGTH`	The distance between the plane and the Glide beacon.	Feet	
`NAV GSI`	Glideslope needle deflection (+/- 119). Note that this provides only 8 bit precision, whereas NAV GLIDE SLOPE ERROR provides 32 bit floating point precision.	Number	
`NAV GS FLAG`	Glideslope flag.	Bool	
`NAV GS LATLONALT:index`	Returns the glide slope.	`SIMCONNECT_DATA_LATLONALT` structure	
`NAV GS LLAF64`	Nav GS latitude, longitude, altitude.	`SIMCONNECT_DATA_LATLONALT` structure	
`NAV HAS CLOSE DME`	Flag if found a close station with a DME.	Bool	
`NAV HAS CLOSE LOCALIZER`	Flag if found a close localizer station.	Bool	
`NAV HAS DME`	Flag if tuned station has a DME.	Bool	
`NAV HAS GLIDE SLOPE`	Flag if tuned station has a glideslope.	Bool	

`NAV HAS LOCALIZER`	Flag if tuned station is a localizer.	Bool	
`NAV HAS NAV`	Flag if Nav has signal.	Bool	
`NAV HAS TACAN`	Flag if Nav has a *Tacan*.	Bool	
`NAV IDENT`	ICAO code.	String	
`NAV LOCALIZER`	Localizer course heading.	Degrees	
`NAV LOC AIRPORT IDENT`	The airport ICAO ident for the localizer that is currently tuned on the nav radio (like 'EGLL' or 'KJFK')	String	
`NAV LOC RUNWAY DESIGNATOR`	The letter code for the runway that the currently tuned localizer is tuned to.	String 'L' - Left 'R' - Right 'C' - Center 'W' - Water 'A' - A 'B' - B	
`NAV LOC RUNWAY NUMBER`	NAV LOC RUNWAY NUMBER - The number portion of the runway that the currently tuned localizer is tuned to (so if the runway was 15L, this would be 15).	String '1' - '36' 'N' 'NE' 'E' 'SE' 'S' 'SW' 'W' 'NW'	
`NAV MAGVAR`	Magnetic variation of tuned Nav station.	Degrees	
`NAV NAME`	Descriptive name.	String	
`NAV OBS`	OBS setting. Index of 1 or 2.	Degrees	

`NAV RADIAL`	Radial that aircraft is on.	Degrees	
`NAV RADIAL ERROR`	Difference between current radial and OBS tuned radial.	Degrees	
`NAV RAW GLIDE SLOPE`	The glide slope angle.	Degrees	
`NAV RELATIVE BEARING TO STATION`	Relative bearing to station.	Degrees	
`NAV SIGNAL`	Nav signal strength.	Number	
`NAV SOUND:index`	Nav audio flag. Index of 1 or 2.	Bool	
`NAV STANDBY FREQUENCY:index`	Nav standby frequency. Index is 1 or 2.	MHz	
`NAV TOFROM`	Returns whether the Nav is going to or from the current radial (or is off).	Enum: 0 = Off 1 = TO 2 = FROM	
`NAV VOLUME`	The volume of the Nav radio.	Percent	
`NAV VOR DISTANCE`	Distance of the VOR beacon.	Meters	
`NAV VOR LATLONALT:index`	Returns the VOR station latitude, longitude and altitude.	`SIMCONNECT_DATA_LATLONALT` structure	
`NAV VOR LLAF64`	Nav VOR latitude, longitude, altitude.	`SIMCONNECT_DATA_LATLONALT` structure	

TACAN

The table below lists all the SimVars related to the *Tacan* system on an aircraft. You can have either 1 or 2 of these systems, and in so all these

SimVars require an index value of 1 or 2, depending on which one you want to target. Note that you must have first enabled the Tacan using the `Tacan.N` parameter in the [systems.cfg](#) file.

Simulation Variable	Description	Units	Settable
`TACAN ACTIVE CHANNEL:index`	The active channel used by the indexed *Tacan* receiver on the aircraft, from 1 to 127.	Number	
`TACAN ACTIVE MODE:index`	The active mode used by the indexed *Tacan* receiver on the aircraft, where 0 = *X* and 1 = *Y*.	Bool	
`TACAN AVAILABLE:index`	Will be TRUE (1) if NAV1, NAV2, NAV3 or NAV4 can receive *Tacan* (depending on the index - 1, 2, 3, or 4), or FALSE (0) otherwise.	Bool	
`TACAN DRIVES NAV1:index`	Tells whether the *Tacan* is driving the Nav 1 indicator (TRUE, 1) or not (FALSE, 0), for autopilot purposes.	Bool	
`TACAN OBS:index`	The *Tacan* OBS setting, in degrees.	Degrees	

`TACAN STANDBY CHANNEL:index`	The standby channel used by the indexed *Tacan* receiver on the aircraft, from 1 to 127.	Number	
`TACAN STANDBY MODE:index`	Indicates the indexed *Tacan* receiver standby mode, where 0 = *X* and 1 = *Y*.	Bool	
`TACAN STATION CDI:index`	The CDI needle deflection amount(course deviation) to the station. Can be +/- 127.	Number	
`TACAN STATION DISTANCE:index`	The distance between the *Tacan* station position and the aircraft position. The index value refers to the Tacan receiver connected to the station (1 or 2).	Meter	
`TACAN STATION IDENT:index`	The tuned station identifier for the indexed *Tacan*.	String	
`TACAN STATION LATLONALT:index`	Retrieves the latitude, longitude and altitude of the *Tacan* station.	`SIMCONNECT_DATA_LATLONALT` structure	
`TACAN STATION RADIAL:index`	The radial between the *Tacan* station and the aircraft.	Degrees	

Simulation Variable	Description	Units	Settable
`TACAN STATION RADIAL ERROR:index`	Difference between the current radial and OBS tuned radial, in degrees.	Degrees.	
`TACAN STATION TOFROM:index`	Returns whether the indexed *Tacan* is going *to* or *from* the current radial (or is off).	Enum: 0 = Off 1 = TO 2 = FROM	
`TACAN VOLUME:index`	The volume value of the indexed *Tacan* receiver on the aircraft.	*Percent Over 100*	

Misc.

Simulation Variable	Description	Units	Settable
`COPILOT TRANSMITTER TYPE`	On which channel the copilot is transmitting.	Enum: 0 = COM1 1 = COM2 2 = COM3 3 = TEL 4 = NONE	
`COPILOT TRANSMITTING`	Whether or not the copilot is transmitting.	Bool	
`PILOT TRANSMITTER TYPE`	On which channel the pilot is transmitting.	Enum: 0 = COM1 1 = COM2	

		2 = COM3	
		3 = TEL	
		4 = NONE	
`PILOT TRANSMITTING`	Whether or not the pilot is transmitting.	Bool	
`RADIOS AVAILABLE`	*Currently not used within the simulation.*	-	
`RADIO HEIGHT`	Radar altitude.	Feet	