



RXP GNS 530W/430W V2

User's Manual For Prepar3d / Flight Sim

Fly Waas Approaches!

The WAAS-certified GNS 530W and its slightly smaller sibling, GNS 430W, lead the industry with multitasking, integrated avionics and cutting-edge WAAS navigation. The GNS 530W comes with optional Class B TAWS alerting to warn you of potential terrain and obstacle conflicts along your flight path. Both come with optional Ryan 9900B TCAD, Shadin Fuel and Shadin Air input, and cross fill capability.

The Reality XP GNS 530W/430W V2 is an entirely new GNS WAAS simulation product built on our unique expertise in designing the most authentic Garmin devices simulation since 2002.

About This Manual

This manual guides you through the product features and is intended for flight simulation purposes only. It shall not be used for any real world aviation application or reference. By reading this manual, you should become well acquainted with the product, and should be able to obtain the information necessary to use the product in your simulator.

We thank you for having chosen a Reality XP Product and wish you a pleasant and a safe virtual flight with us.

Product disclaimer: this software is designed for entertainment only. Although we have designed the product to resemble and function like the original, it is not designed as a training device. Not all systems have been simulated, and some of those that have been simulated may not be entirely functional.

NOT FOR USE IN REAL FLIGHT OR AIRCRAFT OPERATION.

INTRODUCTION	3
System Requirements	3
GETTING STARTED	4
Garmin Reference Guides	4
Add-Ons Menu	4
Configuration Panel	5
Using The Mouse	5
Resizing The Gauge Window	5
ADD-ON DETAILS	6
Autopilot	6
Radios	6
Traffic Collision Alerting Device	6
Taws-B	6
Integral Lighting	6
Cross Fill	7
Flight Simulator Gps Commands	7
Keyboard Shortcuts	7
Discrete Inputs	7
Additional Gauges	7
Failures Annunciation	8
Hardware Controllers	8
Flight Plans And User Waypoints	9
CONFIGURATION	10
Aircraft Configuration	10
Advanced Settings	10
Gauge Rendering Settings	11
Third Party Vendor Settings	11
Keyboard Shortcuts	12
Custom Sim Variables	13
Custom Gps Commands	14
Configuring Alternate Paths	16
Configuring Aviation Database File	16
Installing Multiple Garmin Trainers	16
Device Embedded Memory	17
TROUBLESHOOTING	18

Introduction

The Reality XP GNS 530W and GNS 430W V2 gauges is a faithful reproduction that pilots and flight simulator users can use it as a training tool to familiarize themselves with the workings of the actual equipment. Each button and knob is fully functional and performs identically to its real-world counterpart.

Garmin avionics have become the clear #1 choice in the owner-flown aircraft market. Nearly three-fourths of all U.S. single- and twin-engine piston and turbine aircraft retrofitted since 2000 are now flying with Garmin GPS-enabled equipment. The provision of the integrated communication and navigation radios in the unit gives the extra capabilities of auto tuning and pre-fetching your navigation and approaches frequencies. Flying has never been easier!

The Reality XP GNS 530W and GNS 430W V2 are navigator's heaven with an extraordinary number of user customizable navigation information fields. In short, there is simply no better simulation of the GNS 530W and GNS 430W for any flight simulator, period.



System Requirements

The Reality XP GNS 530W/430W V2 requires either **Prepar3d v4/3/2/1**, **Flight Simulator X - Steam Edition**, **Flight Simulator X** or **Flight Simulator 2004** for Windows¹.

During installation you will be prompted to download and install the Garmin GNS Trainer Lite². The first time you install this package, or should you need to repair the Garmin Trainer installation, the installer will save a copy of the trainer installer in: "C:\ProgramData\Reality XP\Common\GnsTrainer"

This Garmin trainer is a more recent version never offered before in flight simulation, exclusively showcased in our solution. It has main software version 3.3 released in 2010 and includes bug fixes, faster updates, takes less resources and adds some new capabilities to the integrated navigator.

¹ Versions required: Prepar3d 4 (64bits), P3D3, P3D2, P3D1, FSX-SE, FSX-ACC/SP2, FS9-SP1; Windows versions required: WinXP to Win10

² Compatible trainer versions: 3.3

Getting Started

Garmin Reference Guides

In addition to this user's manual which guide you through the functions within the simulator, we recommend referring to the original Garmin GNS series pilots and cockpit reference guide that is installed in its own Garmin program group. Please take the time to read all manuals completely so that you can become properly acquainted with the product and its operation.

Add-Ons Menu

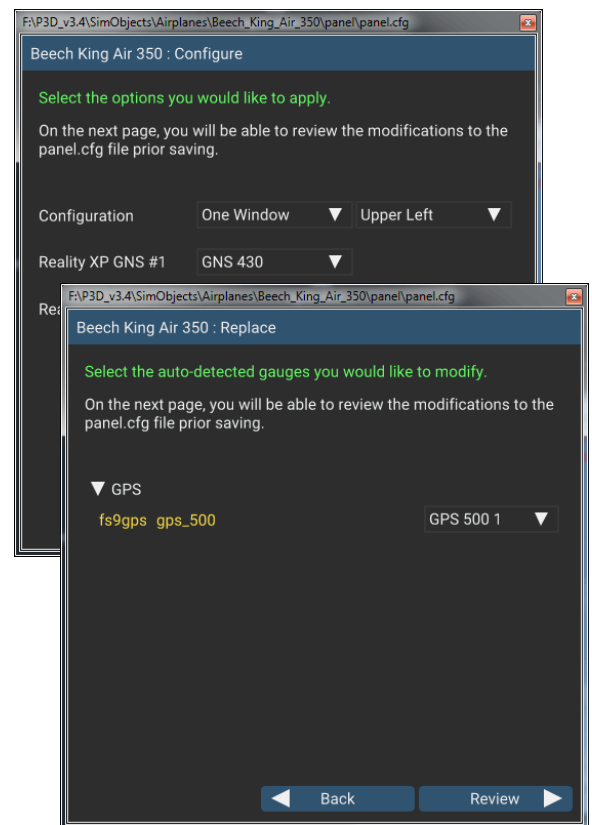
The GNS 530W/430W V2 Menu loads automatically and is operated using the Prepar3d/FSim add-ons menu³.

Select the **menu Add-ons | Reality XP GNS** to add or remove GNS units to your aircraft panel. Once selected, the panel wizard opens and guides you through. This task-oriented context aware assistant can help you with the following tasks:

- Configure your aircraft panel with GNS popup windows.
- Replace any similar GPS gauge with a GNS gauge using an auto-detect feature.
- Remove all GNS gauges from the panel.
- Restore the panel to its prior state using the automatic backup.

The first time the assistant modifies the panel, it saves a backup copy of the panel.cfg file next to the original in the same folder. The last option restores the panel from this backup.

Should you manually modify the panel.cfg file afterward, make sure to make your own copy prior using the 'Restore' function otherwise the manual modifications will be lost.



³ FS9 only: the menu displays in a popup when right-clicking anywhere on the FS9 window. See Troubleshooting chapter for other simulators.

Configuration Panel

The GNS can be tailored to suit your needs by configuring the way each device looks and behaves. Most of each device configuration options are available from the configuration panel.

Press and hold the SHIFT key then RIGHT-Click the top edge of the GNS device to open the configuration panel.

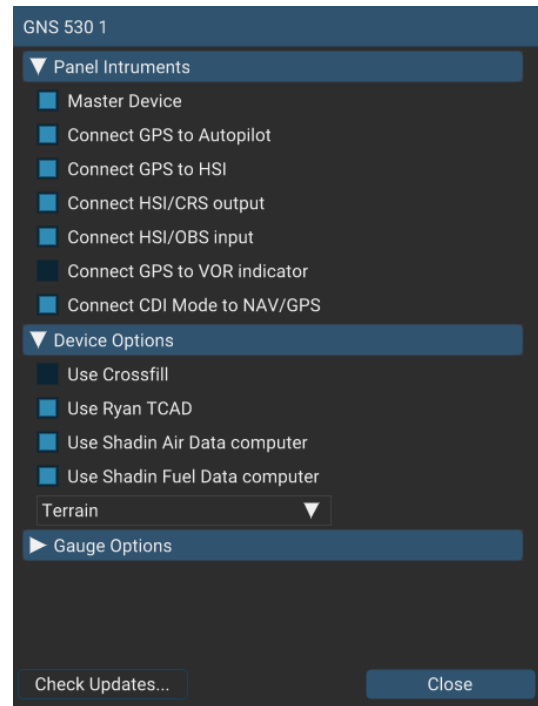
Options are set up for an individual aircraft and then saved to the "RealityXP.GNS.ini" file in the aircraft folder. The add-on restores the last known settings when loading the aircraft.

Whenever the configuration changes, the last row changes to display 'Cancel' and 'Accept' buttons.

When changed settings require rebooting the device, a warning 'reboot' tag displays next to each one and the confirmation button changes to 'Accept and Reboot'.

Select **Check Updates...** to check if you are running the latest version.

You can also **Drag and Drop** any RealityXP.GNS.ini file onto the panel window to import another aircraft settings.



Using The Mouse

This gauge implements a relatively unique implementation of click spots. They work as follows:

- The mouse cursor contextually changes depending on the possible actions.
- As the mouse cursor passes over a click spot it changes from an arrow cursor to a “hand” cursor.
- Whenever a single click spot is used, a left click will accomplish the task.
- Left clicks turn knobs to the counter-clockwise direction when the mouse is in an area to the left of the knob and to the clockwise direction when the mouse is in an area to the right of the knob.
- Mouse wheel turns knobs in either direction regardless if the mouse is to the left or to the right.

Resizing The Gauge Window

The simulation window resizes and maintains its aspect ratio. **Press and hold the SHIFT key then LEFT-Click the top edge** of the GNS device to scale the window to either 1:1 or 2:1 screen sizes.

Add-on Details

The simulation installs as a plug-in in "My Documents\Prepar3d v4 Add-ons\Reality XP GNS" folder⁴. There is only one plug-in for both the GNS 530W and the GNS 430W simulation⁵.

In addition to GPS integration to the panel system, the GNS offers some additional features described below.

Autopilot

The GNS is capable of precise course deviation and roll steering outputs that can be coupled to the simulator autopilot so that IFR flight procedures may be flown automatically. It controls the simulator A/P system both horizontally and vertically.

Radios

The GNS 530W and GNS 430W V2 can control the simulator COM and NAV radios.

To accommodate a wider variety of aircraft panels, the GNS 530W license unlocks the **GPS 500W**. Similar in form and function to the 530W, the radio-less sibling fits panels already equipped with radios. It also fits in any panel equipped with the default FS GPS. The GNS 430W license includes the **GPS 400W** and the **GNC 420W**.



Traffic Collision Alerting Device

The GNS offers a faithful simulation of the Ryan 9900B TCAD (Traffic Collision Alerting Device). Internally powered by a complete TCAS II Mark 7 simulation passing the 250 TSIM tests, it comes integrated to the RXP GNS 530W and the RXP GNS 430W V2. Take full control of the TCAD detection shields, view nearby traffic and take actions upon visual and aural TA advisories directly on the GNS device. The TCAD simulation data source is the Flight Simulator AI Traffic.

TAWS-B

The GNS V2 offers both TAWS-B and TERR Proximity modes. Proper integration and use in TAWS-B mode require advanced settings. The GNS also connects to the flaps and the gear in order to adjust its detection sensitivity and operating limits depending on the phase of flight.

Integral Lighting

Using the configuration panel, adjust both the screen and the bezel brightness independently.

⁴ P3D3: "Prepar3d v3 Add-ons"; P3D2, P3D1, FSX-SE, FSX, FS9: in their respective "Gauges" folder.

⁵ The same gauge file installs both but it requires purchasing and installing both products to unlock both GNS models.

Cross Fill

When enabled in both GNS, this option allows you to transfer a direct- to destination, the active flight plan including VNAV parameters, any stored flight plan or user waypoints to a second 400W or 500W device.

If both units are set to automatic (in the Garmin GNS device settings), a change in the active flight plan, or VNAV parameters, on one unit can also be seen in the other. Initiating a direct-to course to a waypoint on one unit also initiates a direct-to course to the same waypoint on the other unit.

Flight Simulator GPS Commands

The GNS can be configured to intercept all of the default FS GPS commands. When the option is enabled in the configuration panel, the FS GPS bezel buttons and knobs control the GNS.

These commands are described in more details in the 'Configuration' chapter.

Keyboard Shortcuts

Each device supports mapping their buttons and knobs to keyboard shortcuts. These commands are described in more details in the 'Configuration' chapter.

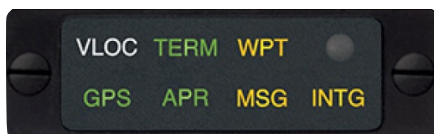
Discrete Inputs

Each device supports additional external controls, either through the keyboard shortcuts or the default GPS commands. These controls change a GNS operating mode (discrete input in a real GNS):

- Toggle CDI source
- Toggle OBS mode
- Inhibit TAWS

Additional Gauges

In addition to the main GNS gauges, the product installs additional **Mid-Continent gauges** ready to be added into any panel⁶. These are fully working replicas displaying valuable information on the panel and including a convenient and functional **TERR INHB** button.



MD41-1484W Annunciator Control Unit



MD41-1028 TAWS Control Unit

⁶ Ready-to-use with the pre-installed add-on.xml on P3D4 and P3D3.

See "C:\Program Files (x86)\Reality XP\Gauges\GPS Annunciation\rxpGpsAnnun (readme).rtf" for more information.

Failures Annunciation

When enabled in the configuration panel, the GNS plugin uses the following sim variables⁷ as failures sources:

rxp.failures.gps_1 rxp.failures.gps_2	Satellite reception failure, displays: "No GPS Position"
rxp.failures.waas_1 rxp.failures.waas_2	Differential GPS failure, reduces location accuracy: satellite reception mode switches to 3D only.
rxp.failures.raim_1 rxp.failures.raim_2	RAIM check failure.
PARTIAL PANEL AVIONICS	Shuts down unit's power.



Hardware Controllers

The GNS supports a selection of external hardware controllers⁸. When the add-on loads, it automatically detects the connected hardware and saves the bindings to the "RealityXP.GNS.ini" file in the aircraft folder. The following hardware devices are supported:

- PFC 430 Control Head Standalone.
- PFC 430 Control Head included in PFC Avionics Stack.

⁷ Variables starting with 'rxp' are additions that can be set from either XML or C++ gauges like any other 'L' var.

⁸ Requires separate activation or a hardware enabled version. See reality-xp.com for available options. Existing customers of the GNS Driver don't have to purchase the driver a second time.

Flight Plans And User Waypoints

Flight plans can be created on a computer using compatible flight planning software and saved to a virtual datacard folder to be imported into the GNS. The imported flight plans can then be stored to the GNS flight plan catalog. This chapter describes how to enable and use this feature.

Preparing the virtual datacard

The first step is to create a virtual datacard folder where to store the flight plans and the user waypoints. This folder must be named **FPL** and be located in the GNS Trainer Data folder, typically⁹:

"C:\ProgramData\Garmin\GNS Trainer Data\GNS\FPL"

The second step is to create one or more flight plan files with the ".fpl" file extension. The product comes with a starter pack installed into the following folder:

"C:\ProgramData\Reality XP\Common\GnsTrainer\flightplans.zip"

Alternatively, this could be a "FPL" shortcut instead of a folder, linked to any other folder on the hard drive, for example:

1. Create a folder "Documents\GNS Flight Plans" where to copy the flight plan files and user waypoints file.
2. Open "C:\ProgramData\Garmin\GNS Trainer Data\GNS\", right-click the empty area to the right, select **new | shortcut**
3. Name the shortcut "FPL" and select the browse button
4. Browse to select the "Documents\GNS Flight Plans" folder and press OK.

Importing flight plans

The last step is to import the flight plans into the GNS device. To proceed with the starter pack, unzip the "flightplans.zip" archive file to extract the "LFATLFRG.fpl" flight plan sample file, then copy the file to the "FPL" folder.

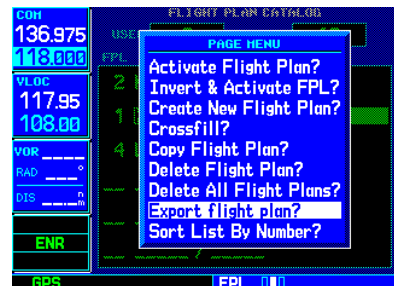
On the GNS device, while viewing the Flight Plan pages, turn the outer GPS knob clockwise to reach the **3rd FPL page** and select flight plan file, and the FPL number this flight plan will be assigned to. When done, select **Import?**



Exporting flight plans

The GNS offers a feature to export the flight plans from the device. On the GNS device, while viewing the Flight Plan Catalog page, press the GPS knob to activate the cursor and select a flight plan in the list.

Then press the **Menu** key and select the **Export Flight Plan?** option. The exported flight plan is saved to the same "FPL" folder with a unique file name.



⁹ The "FPL" folder must reside in the folder equivalent to "GNSAPPPDATA" (see Configuring Alternate Paths chapter for more information).

Configuration

Aircraft Configuration

This chapter details the general organization of the settings file. The GNS settings are automatically saved in the file **RealityXP.GNS.ini** located in the aircraft folder. Every time a GNS opens from the Reality XP GNS menu, it creates this self-documented settings file on save.

The INI file consists of sections and keys. The name of a section is always entered inside the square brackets. Each section contains several keys (the key must be always assigned to the section that begins in the file before this key). The content of one section ends by a definition of the next section. The name of a key is always on a separate line and the content is assigned by the = char. A line comment starts with the ; char.

```
[GNS_530_1]
; is the master device if true.
MasterDevice = true
; selects COM/NAV Radios '1' or '2'.
RadiosPair = 1
; connects GPS to HSI if true.
LinkHsi = true
; connects GPS to HSI course if true.
LinkCrs = true

[GNS_530_1.DEFAULT]
; show screen only gauge if true.
nobezel = false
; enable mouse clickspots if true, disable if false.
usemouse = true
; left mouse button popup ident (ex: GPS_PANEL or 225).
popleft =
```

There are multiple sections in the Reality.GNS.ini file:

- The **GNS_530_1** gauge section contains the settings for this device.
- The **GNS_530_1.DEFAULT** section contains the default settings for each device instance.

Advanced Settings

Advanced settings are only manually editable in the **RealityXP.GNS.ini** settings file in a **[GNS_###_#]** section:

CpuAffinity = -1	selects the CPU running the trainer process: 1 to 32; 0: last core, -1: automatic.
HardwareDevice =	; connects to: 'PFC_STACK', 'PFC_430', 'PFC_530' or any '#PID' (#D002 for PFC_430) no value connects to first found, 'OFF' disables connection (requires the Reality XP GNS Driver add-on)
HardwareIdx = -1	; hardware selector number or -1 to disable (requires the Reality XP GNS Driver add-on)

Gauge Rendering Settings

A Flight Simulator gauge is a template which defines what to display, and the panel.cfg tells where gauges display. Each entry of the same gauge is an instance (a sibling) from the same template (parent). For example, a GNS 530 both in a popup window and the VC Panel is a single gauge with two instances. When adding multiple instances of the same gauge to the panel.cfg file, the **RealityXP.GNS.ini** supports per-instance configuration.

The automatically generated RealityXP.GNS.ini file in the aircraft folder has **[GNS_###_#.DEFAULT]** sections for each device found in the panel. Each gauge sibling uses the default settings from this section, for example [GNS_530_1.DEFAULT]. Should you want a particular instance of the gauge (one sibling) to use other than 'DEFAULT' settings, a new named section must be created and the gauge 5th parameter¹⁰ must be set to this new section name.

For example, to set the VC instance to render without a bezel (screen only): copy and paste the gauge 'DEFAULT' section to the end of the RealityXP.GNS.ini file, change the suffix from 'DEFAULT' to 'NO_BEZEL' then edit the new section settings.

The **RealityXP.GNS.ini** file may be configured like this:

```
[GNS_530_1.NO_BEZEL]
; show screen only gauge if true.
nobezel = true
...
```

The **panel.cfg** file may be configured like this:

```
[VCockpit01]
gauge00=rxpGNS2!GNS_530_1, 0, 0, 100, 90, NO_BEZEL
...
```

In the example above, the 5th param was arbitrarily named 'NO_BEZEL'¹¹. However, you can create as many uniquely named sections as needed, and choose a relevant suffix to match the 5th param:

```
[GNS_###_#.MY_NICE_VC]
[GNS_###_#.MY_COOL_POPUP]
```

Third Party Vendor Settings

Third party aircraft vendor might want to prevent users from altering settings which would break the integration. For example, a specific aircraft integration might require the device to be set to 'COM1/NAV1' radios, and connected to the HSI. The plugin support additional setting to make any entry read-only. When locked, settings appear disable in the configuration panel.

The following key must be manually added in the **RealityXP.GNS.ini** settings file in a **[GNS_###_#]** section:

```
locked = RadiosPair, LinkHsi      comma separated list of read only settings. (NB: case sensitive names)
```

¹⁰ gauge_file ! gauge_name, x, y, w, h, **param**

¹¹ Make sure to use non space separated words.

Keyboard Shortcuts

This chapter describes the list of custom commands available to control the GNS from keyboard shortcuts. Each command exists with the suffix "_1" for unit #1 or "_2" for unit #2. For example, the command name for the device #2 "HOME" button is: "HOME_2".

To change hotkeys, open the "My Documents\Reality XP\GNS Simulation\FitSim\RealityXP.GNS.ini" file with notepad. It is a self-documented configuration file that lets you configure any combination of ALT, CTRL, SHIFT, WIN and key per device button and knob.

CFLP	press button: COM fLIP	PWR_PUSH	press knob: POWER
VFLP	press button: NAV flip	PWR_CCW	rotate knob: POWER (ccw)
CDI	press button: CDI	PWR_CW	rotate knob: POWER (clock-wise)
OBS	press button: OBS	VOL_PUSH	press knob: VOLUME
MSG	press button: MSG	VOL_CCW	rotate knob: VOLUME (ccw)
FPL	press button: FPL	VOL_CW	rotate knob: VOLUME (clock-wise)
VNAV	press button: VNAV	COM_PUSH	press knob: COM
PROC	press button: PROC	COM_OUTR_CCW	rotate outer: COM (ccw)
ENT	press button: ENT	COM_OUTR_CW	rotate outer: COM (clock-wise)
CLR	press button: CLR	COM_INNR_CCW	rotate inner: COM (ccw)
MNU	press button: MNU	COM_INNR_CW	rotate inner: COM (clock-wise)
DTO	press button: DTO	GPS_PUSH	press knob: GPS
RDN	press button: RING UP	GPS_OUTR_CCW	rotate outer: GPS (ccw)
RUP	press button: RING DWN	GPS_OUTR_CW	rotate outer: GPS (clock-wise)
CDI_SOURCE	toggle CDI source (discrete input)	GPS_INNR_CCW	rotate inner: GPS (ccw)
OBS_MODE	toggle OBS mode (discrete input)	GPS_INNR_CW	rotate inner: GPS (clock-wise)
TAWS_INHIBIT	inhibit TAWS (discrete input)	TOGGLE_WINDOW	toggle window visibility

In addition the add-on exports the following additional commands designed to work across multiple Reality XP plugins:

RXP_GPS_MASTER_DEVICE_0	select master device: default GPS
RXP_GPS_MASTER_DEVICE_1	select master device: Reality XP GPS 1
RXP_GPS_MASTER_DEVICE_2	select master device: Reality XP GPS 2
RXP_GPS_MASTER_DEVICE_NEXT	select next available master device

Custom Sim Variables

This chapter describes the list of additional sim variables the GNS publishes to third party plugins¹².

rxp.hsi.bearing_deg_mag_pilot	degrees	
rxp.hsi.flag_from_to_pilot	number	
rxp.hsi.hdef_dots_pilot	number	
rxp.hsi.vdef_dots_pilot	number	
rxp.hsi.has_dme_pilot	number	
rxp.hsi.dme_distance_nm_pilot	miles	
rxp.hsi.dme_speed_kts_pilot	knots	
rxp.hsi.dme_time_sec_pilot	seconds	
rxp.hsi.flag_glideslope_pilot	number	
rxp.hsi.display_horizontal_pilot	number	
rxp.hsi.display_vertical_pilot	number	
rxp.gps.course_degtm	degrees	
rxp.gps.cross_track_nm	miles	
rxp.gps.time_sec_eta	seconds	
rxp.gps.discrete_out	number	
rxp.gps.nav_id	number ¹³	
rxp.radios.indicators.gns_unit1	enum	(4 chars encoded device id or 0) ¹⁴
rxp.radios.indicators.gns_unit2	enum	(4 chars encoded device id or 0)

Control Variables

Third party gauges write to the following sim variables to further enhance the GNS V2 operation:

rxp.panel.rheostat_screen	number	(0 to 100) (reserved for future use)
rxp.panel.rheostat_integral	number	(0 to 100) (reserved for future use)
rxp.failures.gps_1	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.failures.gps_2	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.failures.waas_1	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.failures.waas_2	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.failures.raim_1	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.failures.raim_2	enum	(FAIL_NONE=0, FAIL_INOP=1, FAIL_BLANK=2)
rxp.gps.dscrt_in.obs_mode_sel_1	boolean	GPS #1 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.obs_mode_sel_2	boolean	GPS #2 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.taws_inhib_1	boolean	GPS #1 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.taws_inhib_2	boolean	GPS #2 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.cdi_src_1	boolean	GPS #1 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.cdi_src_2	boolean	GPS #2 discrete in signal (1 to trigger)
rxp.gps.dscrt_in.mute_actv_caut_1	boolean	GPS #1 discrete in signal (1 to trigger) (INOP)
rxp.gps.dscrt_in.mute_actv_caut_2	boolean	GPS #2 discrete in signal (1 to trigger) (INOP)

¹² The "master" device is the only one updating the "rxp.hsi." and the "rxp.gps" variables.

¹³ Because the SDK doesn't allow setting 'string' variables, the ICAO is encoded as a sequence of 6 chars into a 'double' (as binary data).

C code example: `double id = get_named_variable_value(...); const char* szId = (const char*) &id;`

¹⁴ 'G431','G432','G501','G502','G531','G532'; In hex: 0x313333447, 0x323333447, 0x31303547, 0x32303547, 0x313333547, 0x323333547

3D Models Variables

Each device shares the following variables to help synchronizing older 3D models with GNS V2 gauges:¹⁵

<code>rxp.GNS_1.knob_powr</code>	number (rotation angle in degrees)
<code>rxp.GNS_1.knob_volm</code>	
<code>rxp.GNS_1.knob_left</code>	
<code>rxp.GNS_1.knob_rite</code>	

Custom Gps Commands

The GNS intercepts standard Flight Simulator GPS commands with: **Settings Panel | Use Simulator GPS Commands**. These commands are a set of named events which any other gauge, module or 3D model can trigger.

<code>GPS_NEAREST_BUTTON</code>	GNS CDI.
<code>GPS_OBS_BUTTON</code>	GNS OBS.
<code>GPS_MSG_BUTTON</code>	GNS MSG.
<code>GPS_MSG_BUTTON_DOWN</code>	GNS MSG (press).
<code>GPS_MSG_BUTTON_UP</code>	GNS MSG (release).
<code>GPS_FLIGHTPLAN_BUTTON</code>	GNS FPL.
<code>GPS_VNAV_BUTTON</code>	GNS VNAV.
<code>GPS_PROCEDURE_BUTTON</code>	GNS PROC.
<code>GPS_ZOOMIN_BUTTON</code>	GNS zoom in.
<code>GPS_ZOOMOUT_BUTTON</code>	GNS zoom out.
<code>GPS_DIRECTTO_BUTTON</code>	GNS Direct.
<code>GPS_MENU_BUTTON</code>	GNS Menu.
<code>GPS_CLEAR_BUTTON</code>	GNS CLR.
<code>GPS_CLEAR_BUTTON_DOWN</code>	GNS CLR (press).
<code>GPS_CLEAR_BUTTON_UP</code>	GNS CLR (release).
<code>GPS_ENTER_BUTTON</code>	GNS ENT.
<code>GPS_POWER_BUTTON</code>	GNS POWER/VOL Knob.
<code>GPS_ACTIVATE_BUTTON</code>	GNS NAV VOL Knob.
<code>GPS_CURSOR_BUTTON</code>	GNS push cursor.
<code>GPS_GROUP_KNOB_INC</code>	GNS GPS/CRS chapter up (right outer knob cw).
<code>GPS_GROUP_KNOB_DEC</code>	GNS GPS/CRS chapter dn (right outer knob ccw).
<code>GPS_PAGE_KNOB_INC</code>	GNS GPS/CRS page up (right inner knob cw).
<code>GPS_PAGE_KNOB_DEC</code>	GNS GPS/CRS page dn (right inner knob ccw).
<code>GPS_BUTTON1</code>	GNS COM/NAV toggle.
<code>GPS_BUTTON2</code>	GNS COM/NAV coarse up (left outer knob cw).
<code>GPS_BUTTON3</code>	GNS COM/NAV coarse dn (left outer knob ccw).
<code>GPS_BUTTON4</code>	GNS COM/NAV fine up (left inner knob cw).
<code>GPS_BUTTON5</code>	GNS COM/NAV fine dn (left inner knob ccw).
<code>GPS_TERRAIN_BUTTON</code>	GNS TAWS Inhibit (discrete in).

The GNS also supports the following standard Flight Simulator commands:

<code>KEY_COM_RADIO_SWAP</code>	GNS COM FLIP/FLOP.
<code>KEY_COM2_RADIO_SWAP</code>	GNS COM FLIP/FLOP.
<code>KEY_NAV1_RADIO_SWAP</code>	GNS NAV FLIP/FLOP.
<code>KEY_NAV2_RADIO_SWAP</code>	GNS NAV FLIP/FLOP.

¹⁵ Replace `rxp.GNS_1.xxx` with `rxp.GNS_2.xxx` for the 2nd unit variables. The GNS V2 also shares legacy Reality XP GNS vars with: **Settings Panel | Use Legacy 3rd Party Aircraft Panel (L:vars)**

Command Modifiers

Flight Simulator commands have a parameter which is generally only used for actions requiring a value, like the position of a flap lever. Other actions which toggle or trigger a state change usually pass the value 0. The GNS uses this existing parameter in order to modify the "KEY_GPS" (C++) or the "GPS" (XML) command action. This enhances integration between the GNS and 3D cockpit models, add-ons modules, any other XML or C gauges. The command parameter is interpreted as a bit-field where each binary digit represents a specific modifier:

- Bit #2: press and hold the button if set.
- Bit #3: release the button if set.
- Bit #4: command to GPS unit #2 if set, to GPS unit #1 if clear.

Sending the **MSG** key **press-release** event to **GPS#1: GPS_MSG_BUTTON, 0**

Sending the **MSG** key **press-release** event to **GPS#2: GPS_MSG_BUTTON, 8 = bit4 (8)**

Sending the **ENT** key **press and hold** event to **GPS#2: GPS_ENT_BUTTON, 10 = bit2 (2) + bit4 (8)**

Sending the **ENT** key **release** event to **GPS#2: GPS_ENT_BUTTON, 12 = bit3 (4) + bit4 (8)**

Exceptions: GPS_POWER_BUTTON and **GPS_ACTIVATE_BUTTON** commands support "press-release" events only. To command knob rotation, use 'press and hold' for 'TURN LEFT' (CCW) and 'release' for 'TURN RIGHT' (CW).

ModelDef.xml Example

Here is how to implement a typical **<PartInfo>** for the GNS V2 Device #2 "Clear" button¹⁶:

```
<Name>RXP_GNSV2_CLEAR_BUTTON_2</Name>
<MouseRect>
  <Cursor>Hand</Cursor>
  <TooltipText>Clear / Cancel (Press and Hold)</TooltipText>
  <MouseFlags>LeftSingle+LeftRelease+DownRepeat</MouseFlags>
  <CallbackCode>
    (M:Event) 'LeftSingle' scmp 0 == if{ 10 (>K:GPS_CLEAR_BUTTON) }
    (M:Event) 'LeftRelease' scmp 0 == if{ 12 (>K:GPS_CLEAR_BUTTON) }
  </CallbackCode>
</MouseRect>
```

¹⁶ see complete canonical implementation in "C:\Program Files (x86)\Reality XP\GNS Simulation\FitSim\modeldel (RealityXP GNS).xml"

Configuring Alternate Paths

The Garmin trainer installs various components, each one in a specific path. This chapter describes supported Windows environment variables¹⁷ to configure alternate trainer paths in a typical scenario.

Set the general trainer application path, that is where the trainer is installed:

```
GNSSIMPATH="C:\Program Files (x86)\Garmin\GNS Trainer\GNS"
```

Set the application data path, that is where it stores the non volatile memory files (where routes, user waypoints and device settings are saved)¹⁸:

```
GNSAPPDATA="C:\ProgramData\Garmin\GNS Trainer Data\GNS"
```

Set the application database path, that is where to look for all the db files (terrain, obstacle, navigation etc.):

```
GNSNAVDATA="C:\Program Files (x86)\Garmin\GNS Trainer\GNS\DB"
```

Configuring Aviation Database File

This chapter describes a procedure when you want or need to use another aviation database file.

Copy the aviation database file to the location of your choice on your hard drive, for example in "My Documents" folder, with a file name "gns_db.bin", then add the following Windows environment variable:

```
GNSDBFILE="C:\Users\MyName\Documents\gns_db.bin"
```

Installing Multiple Garmin Trainers

This chapter describes a procedure when you want or need to install two different versions of the Garmin GNS Trainer on your computer. The Garmin GNS Trainer installer prevents installing two different versions, such as the latest compatible with the Reality XP GNS and an older one. However, the GNS plugin can be configured to use alternate trainer path and data path. Please note the following for dual Trainer installation:

1. Install the trainer for the Reality XP GNS
2. Copy the "GNS" folder located in "C:\Program Files (x86)\Garmin\GNS Trainer\" to the location of your choice, or rename it for example to: "C:\Program Files (x86)\Garmin\GNS Trainer\GNS (3300)\"
3. Add the following Windows environment variables:
name = "GNSSIMPATH" value = *the path to the 'GNS' folder you have chosen above*

for example:

```
GNSSIMPATH="C:\Program Files (x86)\Garmin\GNS Trainer\GNS (3300)\"
```

4. You can now uninstall the trainer and reinstall the old one: the GNS plugin will use the trainer and its data from the location indicated in your environment variables.

¹⁷ how to add environment variables:

<http://www.kunal-chowdhury.com/2016/09/windows-10-anniversary-update-how-to-edit-environment-variables.html>

¹⁸ The GNSAPPDATA default to the path shown If not manually configured otherwise.

Device Embedded Memory

The GNS Trainer stores all user settings in a file similarly to a real device using embedded non-volatile memory. Settings include user preferences such as displayed navigation units, map settings, routes catalog and the user waypoints. They are saved in the following folder:

```
C:\ProgramData\Garmin\GNS Trainer Data\GNS
```

The Reality XP GNS simulation manages individual non-volatile memory file copies, one per-device and per-unit. These are saved in the same folder next to the original file, using the following file name convention:

```
sys_nand0           : default original file.  
sys_nanda, sys_nandb: GNS 5XX #1 and GNS 5XX #2  
sys_nandc, sys_nandd: GNS 4XX #1 and GNS 4XX #2
```

Should you want to backup and restore the settings, you just need to make a copy of the corresponding file(s) listed above to a safe location, then copy them back to the aforementioned folder to restore.

The simulation automatically creates and manages per-device/per-unit nonvol file. In addition, should the original `sys_nand0` file is missing, the GNS Trainer recreates it from scratch. When upgrading from one trainer version to the next, the setting files are upgraded automatically as well.

THIS SPACE INTENTIONALLY LEFT BLANK

Troubleshooting

This chapter describes solutions available to help troubleshooting problems with the add-on and the Garmin Trainer.

Missing Add-On Menu

The Gauge Add-Ons menu requires **SimConnect** which is normally installed automatically with the simulator. For 32 bits simulators, it requires a minimum of SimConnect FSX-SP2. The installer file, **SimConnect.msi** is located on the original installation disks (FSX-ACC/SP2) or in the following folders:

FSX-SE:

SteamLibrary\steamapps\common\FSX\SDK\Core Utilities Kit\SimConnect SDK\LegacyInterfaces\FSX-XPACK

Prepar3D:

P3D\redist\interface\FSX-SP2-XPACK\retail\lib\SimConnect.msi

Should the menu fails to display, you can create or manually edit the add-on/SimConnect files.

P3D4: Start Windows notepad, create a new text document, copy and paste the following content, then save to¹⁹:

"My Documents**Prepar3D v4 Add-ons**\Reality XP GNS\add-on.xml"

```
<?xml version="1.0" encoding="UTF-8"?>
<SimBase.Document Type="AddOnXml" version="3,3" id="add-on">
  <AddOn.Name>Reality XP GNS</AddOn.Name>
  <AddOn.Description>Legendary Gauges Redefined</AddOn.Description>
  <AddOn.Component>
    <Category>DLL</Category>
    <Name>Reality XP GNS Menu</Name>
    <Path>C:\Program Files (x86)\Reality XP\GNS Simulation\FltSim\64\rxpGNS2_menu.dll</Path>
  </AddOn.Component>
  <AddOn.Component>
    <Category>Gauges</Category>
    <Name>Reality XP GNS Gauge</Name>
    <Path>C:\Program Files (x86)\Reality XP\GNS Simulation\FltSim\64</Path>
  </AddOn.Component>
</SimBase.Document>
```

P3D3: Do as above but change both <Path> lines '64' to '32' as shown below, and save the file to:

"My Documents**Prepar3D v3 Add-ons**\Reality XP GNS\add-on.xml"

```
<Path>C:\Program Files (x86)\Reality XP\GNS Simulation\FltSim\32\rxpGNS2_menu.dll</Path>
<Path>C:\Program Files (x86)\Reality XP\GNS Simulation\FltSim\32</Path>
```

¹⁹ make sure the file extension is .xml and not .txt

P3D2, P3D1, FSX-SE, FSX: The menu uses the global DLL.xml file found here:

"C:\Users\[name]\AppData\Roaming\Microsoft\FSX\dll.xml"

Open the file with notepad, then copy and paste the following **<Launch.Addon>** **</Launch.Addon>** section²⁰:

```
<?xml version="1.0" encoding="UTF-8"?>
<SimBase.Document Type="Launch" version="1,0">
  <Descr>Launch</Descr>
  <Filename>dll.xml</Filename>
  <Disabled>False</Disabled>
  <Launch.ManualLoad>False</Launch.ManualLoad>
  <Launch.Addon>
    ...
  </Launch.Addon>
  <Launch.Addon>
    <Name>Reality XP GNS Menu</Name>
    <Disabled>False</Disabled>
    <ManualLoad>False</ManualLoad>
    <Path>C:\Program Files (x86)\Reality XP\GNS Simulation\FltSim\32\rxpGNS2_menu.dll</Path>
  </Launch.Addon>
</SimBase.Document>
```

Log Files

The solution monitors certain key operations and logs runtime data to specific files. When troubleshooting a problem, the first step is to analyze runtime data saved to the log files.

The gauge log file: "**My Documents\rxpGNS2.gau.log**"

The GNS Simulation log file: "**My Document\rxpGnsSim.dll.log**"

The add-on menu log file: "**My Documents\rxpGNS2_menu.dll.log**"

Debug Settings

The GNS simulation support additional settings to help troubleshooting video card drivers, and to report internal Garmin trainer errors. When enabled, the trainer will log additional information to "rxpGnsSim.dll.log"

The following keys must be manually added in the **RealityXP.GNS.ini** settings file in a **[GNS_###_#]** section:

LogTrainer = 0	1 to enable Garmin Trainer logs.
----------------	----------------------------------

²⁰ Everything included within **<Launch.Addon>** and **</Launch.Addon>** is GNS specific, the rest is the standard DLL.xml file content.

APPENDIX A: Configuration Panel Settings File - RealityXP.GNS.ini file default values²¹:

```
[GNS_530_1]
; is the master device if true.
MasterDevice = true
; selects COM/NAV Radios '1' or '2'.
RadiosPair = 1
; uses computer time if true, simulator time otherwise.
ComputerTime = false
; integral lighting percent (0 to 100)
Rheostat.Display = 100
; integral lighting percent (0 to 100)
Rheostat.Buttons = 75
; connects GPS to HSI if true.
LinkHsi = true
; connects GPS to HSI course if true.
LinkCrs = false
; connects HSI OBS to GPS if true.
LinkObs = false
; connects GPS to VOR if true.
LinkVor = false
; connects GPS to Autopilot if true.
LinkOto = true
; connects CDI key to NAV/GPS switch if true.
AutoNavGps = true
; selects VLOC NAV '1' or '2'. '0' selects same source as 'RadiosPair'.
AutoNavSource = 0
; sets device power source: ALWAYS_ON,AVIONICS_BUS,MAIN_BATTERY
PowerSource = AVIONICS_BUS
; connects GPS to Shadin Air Data computer if true.
UseAirData = true
; connects GPS to Shadin Fuel Data computer if true.
UseFuelData = true
; connects GPS to RYAN 9900BX if true.
UseRyanTcad = false
; sets TCAD sound alerts volume in percent: 0 to 100.
RyanTcadVolume = 50
; connects GPS #1 to GPS #2 if true (both required).
UseCrossFill = false
; enable device failures if true.
UseFailures = false
; uses Simulator GPS commands if true.
UseSimGpsCmds = false
; uses legacy GNS WAAS 3rd party aircraft panel variables (L:Vars).
UseLegacyVars = false
; sets fuel type: AVGAS,JETA,JETB
FuelType = AVGAS
; sets TAWS type: TERR,TAWS
TawsMode = TERR
; sets TAWS surface: HARD,HARD_SOFT,WATER,ANY
TawsSurface = HARD_SOFT
; TAWS minimum runway length (feet)
TawsMinLength = 40
; HSI lateral deviation scale factor.
CdiScale = 1
; HSI vertical deviation scale factor.
```

²¹ The GNS V2 configuration panel stores the settings in the aircraft folder.

REALITY XP GNS 530W/430W V2 P3D/FSIM

```
GsiScale = 1
; sets master sound volume in percent: 0 to 100.
MasterVolume = 100
; selects the CPU running the trainer process: 1 to 32
; 0: last core, -1: automatic.
CpuAffinity = -1
; connects to: 'PFC_STACK','PFC_430','PFC_530' or any '#PID' (#D002 for PFC_430)
; no value connects to first found, 'OFF' disables connection.
HardwareDevice =
; hardware selector number or -1 to disable.
HardwareIdx = -1

[GNS_530_1.DEFAULT]
; show screen only gauge if true.
nobezel = false
; screen only border size (pixels).
border.size = 0
; screen only border color (#RGB or #RGBA).
border.rgb = #000000
; display mouse tooltips if true.
tooltips = false
; enable mouse clickspots if true, disable if false.
usemouse = true
; enable alternate click-spots (left CCW, right CW, middle Push) if true.
usealtmouse = false
; auto-resize dimension (width,height)
refsize =
; left mouse button popup ident (ex: GPS_PANEL or 225).
popleft =
; right mouse button popup ident (ex: GPS_PANEL or 225).
popright =
; adjust brightness (0 to 100)
brightness.bezel = 100
; offset brightness (-100 to +100)
brightness.screen = 0
```

This manual is intended for Flight Simulation use only, and may not be used in any real world aviation applications. The authors are not responsible for any errors or omissions. The information and data given in this user manual can be changed without prior notice. No part of this manual may be reproduced in any form or by any means without the written permission of the publisher, regardless of which method or which instruments, electronic or mechanical, are used. All technical information, drafts, etc. are liable to law of copyright protection.

Inclusion of Garmin copyrighted material in this presentation does not imply any endorsement by Garmin Ltd or its affiliates of the flight training material provided by Reality XP. Please note that Garmin Ltd. or its affiliates owns the copyright to this material and it is reproduced by permission.

©2002-2018 Reality XP all rights reserved.