**EZPushback 0.3.2 (beta)**

Copyright © 2013 Rich Lucas (thePuffyPuff)

**Synopsis**

EZPushback is a lightweight pushback utility designed for X-Plane 10.20 or later. It simply moves the aircraft backward or forward. It can be controlled by menu, keyboard, joystick, or from other plugins. EZPushback is a pure C plugin with no dependencies (i.e. Python, Gizmo, OpenSceneryX, etc.). It is open source and MIT licensed, so hopefully it can be maintained (by me or others) for years to come as X-Plane changes and evolves.

**Requirements**

* X-Plane 10.20 or later (32 or 64 bit version)
* Windows or Mac OSX (Linux not yet supported)

Linux users: I do not have X-Plane development environment for Linux at this time. I will gladly port to Linux if the initial reviews for Windows and MAC versions are favorable and after the beta version is stable.

**Beta**

This is beta software. The following environment was used for build and test.

* Windows 7 x64
* Mac Mini running OSX 10.8.3
* X-Plane 10.20+ (32 and 64 bit versions)
* EADT x737 and stock 747.

**Installation**

The EZPushback.zip file contains a "FAT plugin" that supports both 32 and 64 bit versions X-Plane.

1) Download and unzip the EZPushback.zip file to a temporary location. At the top level there will be one "EZPushback" folder.

2) Find your X-Plane Resource/plugins folder.

3) Copy or move the EZPushback folder so that the EZpushback folder and contents reside within your X-Plane Resources/plugins folder.

4) Start X-Plane (32 or 64 bit).

5) Mouse over the X-Plane main menu and click "Plugins".

An EZPushback menu will appear on the Plugins menu if successfully installed.

**Usage**

EZPushback provides two services: pushback (reverse) and tug (forward). The EZPushback plugin menu provides one way to request services:

X-Plane Main Menu **>**

Plugins **>**

EZPushback **>**

Start Pushback (reverse)

Start Tug (forward)

--------------------------------

Stop

Pushback or tug will not start if the parking brake is engaged or if the aircraft is already moving. Engines may be started during pushback or tug (this is a common practice by some airlines).

Once moving, you can steer normally (i.e. via rudder pedals). Avoid sharp turns while steering as these may induce skidding. Application of brakes or thrust should obviously be avoided during movement as this will fight against the imaginary pushback truck; these actions are not currently prohibited by EZPushback, but may be in future versions.

A transparent status window appears when an EZPushback command is invoked. If you move the window around, EZPushback will remember the location. If the X-Plane window is resized, the EZPushback status window may reposition itself to ensure that it is visible.

A STOP button will appear in the status window when a pushback or tug is in progress. This button can be clicked to stop a push or tug operation.

**Commands**

In addition to the menu options described above, EZPushback provides five custom commands that can be used to request services. These commands can be assigned to keys, joystick buttons, or invoked from other plugins via the X-Plane plugin API.

|  |  |
| --- | --- |
| **Command** | **Provides** |
| tpp/ezpushback/start\_push | Start pushback operation (move backwards). This operation is not allowed unless the aircraft is stopped and the parking brake released. |
| tpp/ezpushback/start\_tug | Start tug operation (move forwards). This operation is not allowed unless the aircraft is stopped and the parking brake released. |
| tpp/ezpushback/stop | Stop a pushback or tug operation that is in progress. |
| tpp/ezpushback/toggle\_push | This command will toggle between pushback and stop using one command. |
| tpp/ezpushback/toggle\_tug | This command will toggle between tug and stop using one command. |

If you are not familiar with keyboard and joystick assignments in X-Plane, please check out Sandy Barbour's video: <http://www.youtube.com/watch?v=LvFZXzq6gQ0>

**Configuration File**

The EZPushback plugin directory contains a configuration file named ezpushback.ini. You may change acceleration, deceleration and target\_speed by editing this file. The x and y values will be overwritten by the plugin when the status window is moved.

|  |
| --- |
| # ezpushback.ini  # You can change the values in this file.  # This file will be written and reformatted when the plugin saves state.  # dialog window position (x=left, y=top)  # moving the window in the sim will change x and y upon save  x=760  y=1033  # acceleration/deceleration in meters per second per second  acceleration=0.20  deceleration=0.50  # target speed in meters per second  target\_speed=1.00 |

**Datarefs**

EZPushback provides several custom datarefs that may be used to monitor and control EZPushback from other plugins. You can also use the datarefs to monitor EZPushback using Dataref Editor. Note that there are no datarefs provided to start pushback, start tug, or stop operations; these functions can be invoked from other plugins by using the same custom commands documented above.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataref** | **Data type** | **read/write** | **Description** |
| tpp/ezpushback/state | int | r | Current state:  0 = Idle  1 = Waiting (not currently used)  2 = Starting (accelerating)  3 = Active (moving at target speed)  4 = Stopping (decelerating) |
| tpp/ezpushback/direction | int | r | Direction of travel:  -1 = Reverse (pushback)  0 = Stopped  1 = Forward (tug) |
| tpp/ezpushback/flags | int | r | Error flags (bit field). Must be 0 before pushback or tug is allowed:  0x0000 = OK  0x0001 = Aircraft not loaded  0x0002 = Parking brake engaged  0x0004 = Aircraft already moving |
| tpp/ezpushback/target\_speed | float | rw | Target speed in meters/sec.  Initially set from config file. Changes to this dataref are not written to config file. |
| tpp/ezpushback/acceleration | float | rw | Acceleration in meters/sec/sec applied when pushback or tug operation begins.  Initially set from config file. Changes to this dataref are not written to config file. |
| tpp/ezpushback/deceleration | float | rw | Deceleration in meters/sec/sec applied when pushback or tug operation ends. Initially set from config file. Changes to this dataref are not written to config file. |
| tpp/ezpushback/local\_vx  tpp/ezpushback/local\_vz | float | r | The current velocity (meter/sec) EZPushback is applying to the aircraft in the X and Z direction. This is computed based on the desired target velocity and aircraft true heading. |
| tpp/ezpushback/callbacks | int | r | The number of flight loop callbacks invoked since pushback or tug operation began. 0 when EZPushback is idle. |

**Alternatives**

SimplePushback from Snailpup provides comparable functionality to EZPushback for versions of X-Plane earlier than X-Plane 10.20. EZPushback was written because SimplePushback is no longer maintained and does not work with 64-bit X-Plane versions.

Ground Services from Joan provides animated pushback, fueling, and other robust features. Ground Services has two large dependencies: Python and OpenSceneryX.

**License**

EZPushback

Copyright © 2013 Rich Lucas (thePuffyPuff)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

**Acknowledgements**

Many thanks to Laminar Research for an awesome sim.

Many thanks to the X-Plane community for your contributions, advice and support.

Special thanks to:

* Sandy Barbour (plugin API support and GoFlight plugin)
* Philipp Münzel (universal X-Plane plugin dev on every platform)
* EADT (x737)
* MarRog (XHSI)
* Javier Cortes (UFMC)
* Snailpup (SimplePushback)
* Joan (Ground Services)