PMDG 737

Developed for Microsoft Flight Simulator

Software Development Kit

The PMDG Simulations Web Site can be found at:
http://www.precisionmanuals.com

COPYRIGHT AND LICENSE

The original purchaser of this software is granted a limited license to use the software in accordance with the End User License Agreement as agreed to at the time of installation, or in the case of the SDK at the time the archive is accessed in order to obtain the SDK materials.

Please review the license agreement carefully, as it provides you with only limited rights. ***Specifically, you may not sell, resell, trade or barter this software product/license without the permission of PMDG.***

You should also be aware that you may not use this simulation software for any training, pilot familiarization, recurrent training or operational awareness training.

This software may not be used to demonstrate the airplane, airplane systems, operational norms, flows, procedures or other pilot knowledge application in a classroom or training environment without being supplemented by the appropriate commercial license.

Please note that this version of the simulation may or may not accurately represent the actual operation of many different aircraft systems and no warranty is made to accuracy or correctness.

In all circumstances the aircraft manuals issued by a certified training center for use with a pilot's training course and the manuals located on the flight deck of an aircraft as well as the operational procedures dictated by the aircraft manuals supersede any information taken from this product or the documentation provided with this product.

This simulation may not be used in any public or private display for which users are charged admission, usage fees or other revenue generating charges. Nor may the simulation be used in any manner which reflects poorly on PMDG, PMDG Simulations, Boeing, Boeing's employees, suppliers or customers.

The SDK associated with this software may not be used to create interfaces or devices to be used for commercial training, entertainment or familiarization, nor may any individual or company sell or use an interface created with this SDK for use in commercial training, entertainment or familiarization without the interface also being paired with an SDK Commercial User's License and a **PMDG 737** Commercial User's License provided by PMDG Simulations, LLC.

Developers who wish to sell software that uses this SDK to interface with the **PMDG 737** line of products in an environment outside of the retail entertainment market must first obtain and sign a license agreement with PMDG Simulations, LLC for the commercial use of this SDK in their products.

TABLE OF CONTENTS

PMDG TECHNICAL SUPPORT

This SDK presumes familiarity with standard C++ coding practices and is designed primarily for use by skilled software developers who are seeking to interface with the **PMDG 737** software.

Please understand that PMDG, PMDG Simulations and affiliates are unable to provide coding assistance, advice or support for developers seeking to use this SDK.

This SDK is provided on an "as-is" basis and is not subject to support or revision by PMDG Simulations, LLC.

PMDG 737 SOFTWARE DEVELOPMENT KIT

The purpose of the **PMDG 737** SDK is to allow users a clean, simple methodology for creating third party applications that interface with the **PMDG 737** series of software products.

This SDK is designed primarily with the intent to allow home-cockpit hardware manufacturers to create simple interfaces so that their hardware can be fully compatible with the **PMDG 737** series of products without having to plumb the depths of memory in search of bits.

We anticipate that other uses for this SDK will come forward that we have not yet imagined, and we encourage creative parties to reach out to us if there are specific items that they feel would enhance their ability to provide after-market products to support the **PMDG 737** line of products.

Please understand that PMDG's mission primarily is to provide software to the community and as such we have very limited resources to provide to after-market developers, but we will certainly hear and evaluate any suggestions!

The **PMDG 737** SDK provides an interface which can be used by third party software to monitor the state of the **PMDG 737** and to control certain cockpit functions.

The SDK provides read-only access to a data block containing the state of most **PMDG 737** controls and indications, as well as the contents of the FMC CDU screens. It also allows controlling the **PMDG 737** by triggering control events that request operation of one or more items in the **PMDG 737**.

The communication between the third party add-on application and the **PMDG 737** is done using the SimConnect library that comes standard with Microsoft Flight Simulator. Please refer to the SimConnect SDK documentation for advice on the creation of a SimConnect application.

The **PMDG 737** SDK includes example SimConnect applications written in C++ that you may use as starting points, but they are by no means a complete or final authority on the subject of interfacing with SimConnect.

SDK CONTENTS

The PMDG 737 SDK includes the following files:

- **PMDG_NG3_SDK.h** is the SDK header file. It defines the data structures and events used to communicate with the PMDG 737. You should #include this file in your application.

- **PMDG_NG3_ConnectionTest.cpp** is an example application. It demonstrates how to connect to the **PMDG 737** via SimConnect, read the state of several **PMDG 737** switches and us different methods to send control events to the **PMDG 737*.***

- **PMDG_NG3_SDK_CDU_Test.cpp** is an example application demonstrating how to read the contents of the **PMDG 737*'s*** FMC CDU screens.

ENABLING DATA OUTPUT

The **PMDG 737** is already set up to listen for control events that may originate from an external application. However, since providing state data to third party applications does consume a small amount of processor attention, **we have disabled this communication output by default** since the vast majority of our users will not be using SDK driven applications.

To enable the data communication output, you will need to open the file **737_Options.ini** that is located in the 737 persistent storage folder.
For Microsoft Store distribution, this folder is located at
`%LOCALAPPDATA%\Packages\Microsoft.FlightSimulator_8wekyb3d8bbwe\LocalState\packages\pmdg-aircraft-737\work\`.
For Steam distribution, this folder is located at
`%APPDATA%\Microsoft Flight Simulator\Packages\pmdg-aircraft-737\work\`.
Once this folder is open, add the following lines as required to the bottom of the file:

```
[SDK]
EnableDataBroadcast=1
EnableCDUBroadcast.0=1
EnableCDUBroadcast.1=1
```

Add only those line(s) that enable the data output that you need, such as data or specific CDU units. When you release your product to customers, it will be necessary for your application installation to set these parameters on the customer's installation or else they will not see data connectivity between your application and the **PMDG 737**.

READING CONTROL AND INDICATOR STATES

The **PMDG 737** keeps the state of most of its controls and indicators in a data block that can be read by external applications. (Provided that this capability is enabled, as described on the previous page.)

Your application will use SimConnect to request data from this block. The contents of this data block are defined by `struct PMDG_NG3_Data` in the PMDG_737_SDK.h file.

The members are self-explanatory and we have added comments to clarify wherever the variable is not a simple boolean.

The PMDG_737_ConnectionTest.cpp sample code demonstrates the access to the **PMDG 737**.

For example, the following code from testCommunication() function requests data from the **PMDG 737** data block. The data is sent *whenever it changes:*

```
// Associate an ID with the PMDG data area name
hr = SimConnect_MapClientDataNameToID (hSimConnect, PMDG_NG3_DATA_NAME,
   PMDG_NG3_DATA_ID);

// Define the data area structure - this is a required step
hr = SimConnect_AddToClientDataDefinition (hSimConnect,
   PMDG_NG3_DATA_DEFINITION, 0, sizeof(PMDG_NG3_Data), 0, 0);

// Sign up for notification of data change.
// SIMCONNECT_CLIENT_DATA_REQUEST_FLAG_CHANGED flag asks for the data to be
sent only when some of the data is changed.
hr = SimConnect_RequestClientData(hSimConnect, PMDG_NG3_DATA_ID,
   DATA_REQUEST, PMDG_NG3_DATA_DEFINITION,
   SIMCONNECT_CLIENT_DATA_PERIOD_ON_SET,
   SIMCONNECT_CLIENT_DATA_REQUEST_FLAG_CHANGED, 0, 0, 0);
```

Then, the event dispatch procedure handles the **PMDG 737** state data sent by SimConnect:

```
void CALLBACK MyDispatchProc(SIMCONNECT_RECV* pData, DWORD cbData,
   void *pContext)
{
   switch(pData->dwID)
   {
   case SIMCONNECT_RECV_ID_CLIENT_DATA:
   // Receive and process the NG3 data block
   {
        SIMCONNECT_RECV_CLIENT_DATA *pObjData =
             (SIMCONNECT_RECV_CLIENT_DATA*)pData;

        switch(pObjData->dwRequestID)
        {
        case DATA_REQUEST:
             {
                  PMDG_NG3_Data *pS =
                       (PMDG_NG3_Data*)&pObjData->dwData;
                  ProcessNG3Data(pS);
                  break;
             }
        }
   }
}
```

```
// This function is called when NG3 data changes
void ProcessNG3Data (PMDG_NG3_Data *pS)
{
    // test the data access:
    // get the state of switches and save it for later use
    if (pS->LTS_TaxiSw != NG3_TaxiLightSwitch)
    {
        NG3_TaxiLightSwitch = pS->LTS_TaxiSw;
        if (NG3_TaxiLightSwitch)
            printf("TAXI LIGHTS: [ON]\n");
        else
            printf("TAXI LIGHTS: [OFF]\n");
    }
}
```

READING CDU SCREEN CONTENTS

The SDK also provides data blocks with the contents of the CDU screens. There is one data block for each CDU unit. An external application can request this data by using SimConnect in a way similar to reading the main **PMDG 737** data block discussed earlier.

The contents of the CDU data block are defined by structures PMDG_NG3_CDU_Screen and PMDG_NG3_CDU_Cell in the PMDG_NG3_SDK.h file.

The data includes a 24x14 array of screen cells. Each cell contains the displayed ASCII symbol, the symbol color, and a bit mask value that defines if the symbol is in small font, is shown in reverse video or is dimmed to indicate an unused entry. See the definition of the data structures in the PMDG_NG3_SDK.h for an explanation of each data field. The data block also includes a Boolean value indicating if the CDU unit has power.

The PMDG_NG3_SDK_CDU_Test.cpp sample code demonstrates the access to the left CDU of the PMDG 737.

The sample uses the following code to connect to the **PMDG 737** and request data for the left CDU. The data is sent *whenever it changes*:

```
HRESULT hr;

if (SUCCEEDED(SimConnect_Open(&hSimConnect, "PMDG NG3 CDU Test", NULL, 0,
0, 0)))
{
    // Associate an ID with the PMDG data area name
    hr = SimConnect_MapClientDataNameToID (hSimConnect,
       PMDG_NG3_CDU_0_NAME, PMDG_NG3_CDU_0_ID);

    // Define the data area structure - this is a required step
    hr = SimConnect_AddToClientDataDefinition (hSimConnect,
      PMDG_NG3_CDU_0_DEFINITION, 0,
      sizeof(PMDG_NG3_CDU_Screen), 0, 0);

    // Sign up for notification of data change.
    // SIMCONNECT_CLIENT_DATA_REQUEST_FLAG_CHANGED flag asks for the data
    // to be sent only when some of the data is changed.
    hr = SimConnect_RequestClientData(hSimConnect, PMDG_NG3_CDU_0_ID,
       CDU_DATA_REQUEST, PMDG_NG3_CDU_0_DEFINITION,
       SIMCONNECT_CLIENT_DATA_PERIOD_ON_SET,
       SIMCONNECT_CLIENT_DATA_REQUEST_FLAG_CHANGED, 0, 0, 0);
    }
```

This procedure handles the reception of the CDU data:

```
PMDG_NG3_CDU_Screen screen;

bool checkSimConnect()
{
    SIMCONNECT_RECV* pData;
    DWORD cbData;

    HRESULT hr = SimConnect_GetNextDispatch(hSimConnect, &pData, &cbData);

    if (SUCCEEDED(hr))
    {
        if (pData->dwID == SIMCONNECT_RECV_ID_CLIENT_DATA)
        {
            SIMCONNECT_RECV_CLIENT_DATA *pObjData =
            (SIMCONNECT_RECV_CLIENT_DATA*)pData;

            if (pObjData->dwRequestID == CDU_DATA_REQUEST)
            {
                PMDG_NG3_CDU_Screen *pS =
                (PMDG_NG3_CDU_Screen*)&pObjData->dwData;
                memcpy(&screen, pS, sizeof(PMDG_NG3_CDU_Screen));
                return true;
            }
        }
    }
    return false;
}
```

You can then use code like this to loop through each of the CDU cells:

```
for (int x=0; x<CDU_COLUMNS; ++x)
{
    for (int y=0; y<CDU_ROWS; ++y)
    {
            PMDG_NG3_CDU_Cell *cell = &(screen.Cells[x][y]);  char
            symbol = cell->Symbol;
            bool smallFont = (cell->Flags & PMDG_NG3_CDU_FLAG_SMALL_FONT);
    }
}
```

CONTROLLING THE PMDG 737

Control Event Parameters:

Your application can operate the **PMDG 737** controls by sending special commands. These commands contain both a **control event code** and a **numeric parameter.**

The **control event code** defines the aircraft control, switch, selector or knob that is to be operated. The control events are listed in the PMDG_NG3_SDK.h file.

The **numeric parameter** can be used to send a switch position, or a control parameter to the **PMDG 737**.

Example 1: Sending a value via the **numeric parameter** will tell the simulation to place the switch/knob/lever/control into a specific position. All values below 8192 are treated as a numeric position to which the item being controlled should be placed.

You can determine the positions of various knobs by looking up in the PMDG_NG3_SDK.h file. Boolean parameters have no position information listed, but non Boolean parameters will include position information in the comments following the variable declaration:

```
bool            ELEC_GrdPwrSw;
unsigned char   ELEC_BatSelector;       // 0: OFF  1: BAT  2: ON
```

Example 2: Alternatively the parameter can be one of the following mouse actions:

- MOUSE_FLAG_RIGHTSINGLE
- MOUSE_FLAG_MIDDLESINGLE
- MOUSE_FLAG_LEFTSINGLE
- MOUSE_FLAG_RIGHTDOUBLE
- MOUSE_FLAG_MIDDLEDOUBLE
- MOUSE_FLAG_LEFTDOUBLE
- MOUSE_FLAG_RIGHTDRAG
- MOUSE_FLAG_MIDDLEDRAG
- MOUSE_FLAG_LEFTDRAG
- MOUSE_FLAG_MOVE
- MOUSE_FLAG_DOWN_REPEAT
- MOUSE_FLAG_RIGHTRELEASE

- MOUSE_FLAG_MIDDLERELEASE
- MOUSE_FLAG_LEFTRELEASE
- MOUSE_FLAG_WHEEL_FLIP
- MOUSE_FLAG_WHEEL_SKIP
- MOUSE_FLAG_WHEEL_UP
- MOUSE_FLAG_WHEEL_DOWN

Sending one of these parameters will simulate the associated mouse action being acted upon the switch/knob/control. This technique can be used to rotate knobs or to toggle switches, move levers, etc.

Control Methods:

There are two methods of sending control events to the **PMDG 737**. One involves using the **PMDG 737** data block, and the other involves sending P3D events to the simulation, where they are picked up and processed by the **PMDG 737**.

Which method you choose will depend on your application and preferred methods. Both will yield the same results.

Method 1 uses the special control data area monitored by the **PMDG 737** (This area was described a few pages ago.) The data area is initially empty. To send an event, your application writes the event ID and parameter to this data area. The **PMDG 737** detects non-zero data and processes the corresponding event, after which it zeroes the control data area in order to prepare it for future events.

Note that your application should wait until the command area is zero before placing another command there.

Method 1 example:

The connection to the **PMDG 737** control data area is set up like this:

```
PMDG_NG3_Control Control;
...

Control.Event = 0;
Control.Parameter = 0;

// Associate an ID with the PMDG control area name
hr = SimConnect_MapClientDataNameToID (hSimConnect,
    PMDG_NG3_CONTROL_NAME, PMDG_NG3_CONTROL_ID);
```

```
// Define the control area structure - this is a required step
hr = SimConnect_AddToClientDataDefinition (hSimConnect,
    PMDG_NG3_CONTROL_DEFINITION, 0, sizeof(PMDG_NG3_Control),
0, 0);

// Sign up for notification of control change.
hr = SimConnect_RequestClientData(hSimConnect, PMDG_NG3_CONTROL_ID,
    CONTROL_REQUEST, PMDG_NG3_CONTROL_DEFINITION,
    SIMCONNECT_CLIENT_DATA_PERIOD_ON_SET,
    SIMCONNECT_CLIENT_DATA_REQUEST_FLAG_CHANGED, 0, 0, 0);
```

A code in the dispatch procedure keeps the data synchronized to **PMDG 737**:

```
case CONTROL_REQUEST:
{
// keep the present state of Control area to know if the server
// had received and reset the command
  PMDG_NG3_Control *pS = (PMDG_NG3_Control*)&pObjData->dwData;  Control =
      *pS;
      break;
}
```

This sample code sets the TAXI lights switch:

```
// Send a command only if there is no active command request and
// previous command has been processed by the NG3
if (Control.Event == 0)
{
    Control.Event = EVT_OH_LIGHTS_TAXI;      // = 69749
    if (New_TaxiLightSwitch)
        Control.Parameter = 1;
    else
        Control.Parameter = 0;
    SimConnect_SetClientData (hSimConnect, PMDG_NG3_CONTROL_ID,
        PMDG_NG3_CONTROL_DEFINITION,
        0, 0, sizeof(PMDG_NG3_Control), &Control);
}
```

In this case, the transmitted event is EVT_OH_LIGHTS_TAXI. The available events are listed in the PMDG_NG3_SDK.h file. The control parameter in this case is either 0 or 1 and determines the position the switch should be placed into.

Note how the code checks that the **PMDG 737** has no pending events to process by checking that `Control.Event == 0`.

Method 2 is to directly generate the corresponding P3D event, which will be processed by the **PMDG 737**. The advantage to using this method is that it allows sending one or several events at once without waiting for previous events to be received and processed.

The catch, however, is that it involves defining IDs and using SimConnect_MapClientEventToSimEvent functions for each unique control before you can trigger corresponding events.

Method 2 example:

This is how the direct event triggering is set up:

```
static enum EVENT_ID {
    EVENT_LOGO_LIGHT_SWITCH,
    EVENT_FLIGHT_DIRECTOR_SWITCH
};

...

//EVT_OH_LIGHTS_LOGO
hr = SimConnect_MapClientEventToSimEvent(hSimConnect,
    EVENT_LOGO_LIGHT_SWITCH, "#69754");

//EVT_MCP_FD_SWITCH_L
hr = SimConnect_MapClientEventToSimEvent(hSimConnect,
    EVENT_FLIGHT_DIRECTOR_SWITCH, "#70010");
```

The `"#69754"` is the numerical value for `EVT_OH_LIGHTS_LOGO`, which is defined as

```
#define EVT_OH_LIGHTS_LOGO      (THIRD_PARTY_EVENT_ID_MIN + 122)
```

and `THIRD_PARTY_EVENT_ID_MIN` is equal to 69632.

After this, the logo lights can be switched by triggering the MSFS event:

```
int parameter = 1;      // New switch position
SimConnect_TransmitClientEvent(hSimConnect, 0, EVENT_LOGO_LIGHT_SWITCH,
    parameter, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
    SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
```

Such a call moves the switch to a specified position.

It is also possible to send events that simulate mouse operations. This can be useful to control rotary knobs or to trigger a two position switch without checking its previous position. To do this, set the event parameter to one of MOUSE_FLAG_LEFTSINGLE, MOUSE_FLAG_LEFTRELEASE, etc. (These constants are defined in the PMDG_NG3_SDK.h file.)

The following examples send mouse operation events:

```
// use mouse simulation to toggle the switch
SimConnect_TransmitClientEvent(hSimConnect, 0,
    EVENT_FLIGHT_DIRECTOR_SWITCH, MOUSE_FLAG_LEFTSINGLE,
    SIMCONNECT_GROUP_PRIORITY_HIGHEST,
    SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

SimConnect_TransmitClientEvent(hSimConnect, 0,
    EVENT_FLIGHT_DIRECTOR_SWITCH, MOUSE_FLAG_LEFTRELEASE,
    SIMCONNECT_GROUP_PRIORITY_HIGHEST,
    SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
```

CONCLUSION

Using the information and tools provided by this SDK, a skilled developer should be able to control just about any switch, knob, lever or control within the **PMDG 737** simulation.

In addition, the state of hundreds of other parameters including annunciators and handles can be determined using the same methods described above.

We have made available many more parameters than most developers will likely require, but if your company or organization would like to request that we make others available, please open a support ticket and let us know! We cannot promise we will comply, but we will keep your request on hand for future updates, and we'll see what we can do!

REQUIREMENTS FOR DEVELOPERS USING THIS SDK

The End User License Agreement for the **PMDG 737** SDK is included at the end of this document. It is extremely important that anyone using this SDK read through the EULA carefully, as we have placed some very specific restrictions on what you may do with this SDK and how you may distribute your creations.

To help you understand your responsibilities when using this SDK, some of the key points will be digested here in this section, but this digest is by no means complete. You must read and comply with the full EULA listed below or risk litigation by PMDG Simulations. to protect our property and the limitations we have placed on you.

If you have any questions regarding the use of the application you have created, please contact us! We will be happy to provide you with feedback to keep us all working toward the common goal of better simulation for flight simulation enthusiasts!

One and only Method of Interaction with the* PMDG 737*:

You may only use the **PMDG 737** SDK to interact with the **PMDG 737** Software. You may not create alternate methods of controlling the **PMDG 737** software, nor may you manipulate or otherwise use memory address methods to interact with the **PMDG 737**.

***Display of EULA to users of your* PMDG 737 *SDK created application*:**

You must create an installation and/or authorization method for your creation that displays the **PMDG 737** SDK EULA, and requires the user to accept the terms of the EULA during installation OR at first run-time of the software. The display and acceptance must meet current industry standards and must require the user to manually opt-in to accept the limitations and terms of the **PMDG 737** EULA.

***Free Distribution of SDK Created Works*:**

If you create an application that uses the **PMDG 737** SDK and/or the **PMDG 737** and you intend to distribute your work free of charge, it is important to understand that you are still obligated to comply with the remaining provisions of the EULA.

Non-Commercial Limitation of SDK Created Works:

The works you create with the **PMDG 737** SDK (whether freeware or retail products) may be used only for the private entertainment of individual persons. Your works may not be used by corporations or in a commercial, educational or for-hire environment. Commercial, educational and for-hire environments include but are not limited to: Training centers, Aviation training centers, schools, classrooms, commercial operators that lease and/or sell time to users in cockpit and/or computer setups that simulate the 737 or other aircraft using the **PMDG 737** software or the **PMDG 737** SDK application created by you. This means that if you create an interface or application to connect hardware or software to the **PMDG 737**, it is your responsibility to insure that your application is used only by private individuals and is not used in flight training, flight training centers, flight simulation rental centers/businesses or other businesses engaged in the leasing, loaning or renting of a flight simulation that uses the **PMDG 737** or software created using the **PMDG 737** SDK.

It is incumbent upon you to ensure that any customer or user who employs a tool or utility that you create with the **PMDG 737** *SDK is aware of this limitation and required to accept it prior to installation/use.*

Commercial use of SDK Created Works:

If you would like to create a work to be used in the commercial sphere, it is required that you obtain a commercial application license from PMDG Simulations prior to making your work available to any customer or user, including beta testers. It is important to note here that any commercial application will also require your end customer to purchase a commercial-use license of the **PMDG 737** which can only be acquired from PMDG Simulations. If you are attempting to create a commercial solution and would like to involve the **PMDG 737**, please contact us early in your project so that we can make sure to create a solution that will meet your needs.

***When In Doubt, Reach Out*:**

If you are unclear on any of the requirements described here and/or included in the End User License Agreement, please contact us via our support ticket system at

http://support.pmdg.com

and we will be happy to help clarify how the EULA applies to your specific project!

PMDG 737 & PMDG 737 SDK
END USER LICENSE AGREEMENT

PMDG SIMULATIONS, LLC.
END-USER SOFTWARE LICENSE AGREEMENT

This End-User License Agreement ("EULA") pertains to PMDG Simulations, LLC's Copyrighted computer software (the "Software") in the package containing CD-ROMs, diskettes, associated media, printed materials, electronic distribution files or electronic documentation. Please read the terms and conditions of this End-User License Agreement ("EULA") before installing or using the Software.

This End-User License Agreement ("EULA") is a legally enforceable contract between you ("Customer" or "you") and PMDG Simulations, LLC ("PMDG" or “PMDG Simulations”). By clicking "I agree", installing, copying, or otherwise using any part of the Software or any associated media, any printed materials, or any "online" or electronic documentation, you agree to be bound by the terms of this EULA.

1. GRANT OF NON-EXCLUSIVE, LIMITED LICENSE

The Software and related documentation is licensed, not sold. Any rights not clearly and expressly granted to you under this EULA are reserved to PMDG. Provided you have paid all applicable fees, registered the Software with PMDG, and otherwise complied with this EULA, PMDG Simulations grants you the following personal, non-exclusive, non-transferable, limited rights:

a) One user may use the Software on the Customer's computer where the software was installed; you may use the Software on a second computer so long as only one copy is used at a time and only after having secured a second license permission from PMDG. You may use the Software on a network ONLY IF each individual accessing the Software through the network has a copy licensed to that individual and installed on the machine from which they are accessing the network; and you may copy the Software for archival purposes, provided the copy contains all of the original Software's proprietary notices.

b) This software is for use only with the official versions of Lockheed Martin Prepar3D. You may not use this software on any platform other than Lockheed Martin Prepar3D.

c) You may use the Software for private entertainment purposes only.

d) You may not use this software to conduct or receive training, aircraft familiarization or other training scenarios without first obtaining a commercial support license from PMDG Simulations, LLC.

e) You may not use this software to maintain currency in any aircraft, nor may you use this software to satisfy any certification requirement or training program curriculum.

f) You many not lease, rent or otherwise charge for any person to use the software, a computer running the software, or a device that uses the software without first obtaining a commercial license from PMDG Simulations, LLC.

g) You may not operate this software in a public venue designed to demonstrate hardware or other display, sound, processing or flight simulation components without the express advance permission of PMDG Simulations LLC.

h) You may not charge non-licensed or licensed owners of this software for use of this software, training in conjunction with this software, or while using this software to demonstrate any technique or process that may be considered training or familiarization.

i) You may not alter, reverse engineer or otherwise modify any portion of this software except as described in this license.

j) You shall not have the right to obtain or use any source code for the Software, nor copy, reproduce, disclose, rent, lease, loan, distribute or use the Software except as provided above.

k) If you create an application designed to interact with this software, you may manipulate or control the software using only the specifically provided methods included in the Software Developers Kit provided for use with this product.

l) You may produce user-made liveries using your original artwork and/or using tools provided by PMDG specifically for this purpose. You may alter certain files contained in this program as necessary to activate user-created liveries with this software.

m) You may redistribute royalty-free, distribution fee-free and at no cost any user-created liveries for use with this software, but you may not prevent free, unfettered public access to those livery files through the use of membership, restricted login or other barrier devices that require a fee, cost or charge in order to obtain membership and access.

n) You agree that you will not distribute unauthorized copies of this software to any person by any means.

o) You agree that if you are found to be distributing unauthorized copies of this software, or if you are found to be using a copy of this software that was not purchased from PMDG or an authorized reseller and licensed to you that the damage you will have done to PMDG cannot be reasonably calculated, and thus you agree to cover all applicable legal expense incurred by PMDG in the protection of our rights. You further agree to provide PMDG with $25,000 USD per event above and beyond any reasonable costs related to the protection of our rights.

p) You may not use any part, file or artwork included as part of the PMDG 737 in a product created for use with this or any other product without the specific, written permission of PMDG Simulations, LLC. This includes using the documentation, manuals, art files, liveries, paint kits or any other media included as part of this product.

q) You may not overwrite, replace, edit or otherwise modify any of the files included with the PMDG product without permission from PMDG Simulations, LLC.

r) If you create a freeware or commercial product (except free liveries) that modifies any piece of this product including artwork graphics, configuration files, sound files, ini files, binary files or any other parts of the product as created by PMDG Simulations, LLC. you are obligated to allow PMDG Simulations, LLC. to quality-check your application for unintended consequences in the operation of this software prior to making it generally available to the public.

s) If you are a developer participating in the development of add-on products for Lockheed Martin Prepar3D or any other simulation platform, you may not use this product, the documentation provided, the fonts, files, data or information provided with this product without express written permission of PMDG Simulations, LLC.

t) If you create works using the SDK associated with the software, you must display this EULA to any customer who installs and/or runs your application in conjunction with the software. You must require that the customer signal acceptance of this EULA by manually accepting the agreement at the time it is displayed.

u) You may not, without the advanced permission of PMDG Simulations, LLC. create and distribute work that interacts with the software if that work is to be used in, or will facilitate the use of the software in a commercial training, flight training, familiarization, training, recurrent training, initial training or any other environment in which the software is to be used in violation of this EULA.

v) You may not recommend to any person, institution or organization that they use this software in place of a properly licensed commercial version of this software distributed by PMDG Simulations, LLC. if the application for which the software is to be used would require such a license.

w) Any rights not specifically given to you in the EULA are reserved by PMDG Simulations, LLC.

2. WARRANTY, REMEDY, AND LIMITATIONS

a. PMDG WARRANTS THAT THE SOFTWARE WILL PERFORM IN SUBSTANTIAL ACCORDANCE WITH THE ACCOMPANYING USER MANUAL FOR 90 DAYS FOLLOWING YOUR RECEIPT OF THE SOFTWARE. THIS IS YOUR ONLY WARRANTY.

b. PMDG SIMULATIONS' SOLE OBLIGATION AND YOUR EXCLUSIVE REMEDY FOR ANY WARRANTY FAILURE IS TO FOLLOW PMDG'S REGULAR BUSINESS PRACTICES FOR THE CORRECTION, REPLACEMENT OR REMEDY OF A DEFECT. UNLESS OTHERWISE STATED ABOVE, THE SOFTWARE AND THE DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A ARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THIS SOFTWARE REMAINS WITH YOU. IN NO EVENT WILL PMDG SIMULATIONS OR ANY OF ITS AFFILIATES OR SUPPLIERS BE LIABLE FOR ANY INCIDENTAL, CONSEQUENTIAL, INDIRECT OR OTHER SPECIAL DAMAGES, INCLUDING LOST DATA, LOST REVENUES OR LOST PROFITS, EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. PMDG RESERVES THE RIGHT TO MODIFY THIS DOCUMENT AT ANY TIME WITHOUT OBLIGATION TO NOTIFY ANYONE. IN NO EVENT SHALL PMDG SIMULATIONS' OR ITS SUPPLIERS' LIABILITY UNDER THIS AGREEMENT EXCEED THE SUM OF ANY AMOUNTS PAID HEREUNDER BY YOU TO PMDG SIMULATIONS, LLC. OR THE SUPPLIER. REFUNDS FOR DIGITALLY DELIVERED SOFTWARE WILL BE MADE ONLY IN ACCORDANCE WITH PMDG SIMULATIONS' POLICY AS STATED ON THE PMDG WEB SITE.

REFUND REQUESTS FOR NON DIGITALLY DELIVERED SOFTWARE WILL BE MADE WITH THE RETAILER OR DISTRIBUTOR AND NOT DIRECTLY WITH PMDG SIMULATIONS, LLC.

TERMINATION

The license will terminate automatically if you fail to comply with the terms, conditions, or limitations contained in this EULA, including the payment of applicable license or other fees. You may terminate this EULA at any time (with no obligation on the part of PMDG Simulations, LLC.) by destroying all copies of the Software, deleting any copies of the Software from your hard drives, ceasing all use of the Software and documentation; and providing satisfactory proof to PMDG Simulations, LLC. that you have done so.

The disclaimer of warranty and limitations on liability contained in Section 4 shall continue in force even after your rights to use the Software are terminated.

DEMO OR EVALUATION VERSIONS

If PMDG designates the Software as a "Demo" or "Evaluation" version, you may use the Software solely for evaluation purposes for the designated 10-day period. Use of the Software and/or documentation beyond the 10-day evaluation period violates PMDG's rights, as described above, including but not limited to PMDG's rights under the United States Copyright Act. You acknowledge that any Demo or Evaluation version is merely a technology demonstration that may not be at the level of performance or compatibility of generally available PMDG products. Demo or Evaluation versions are provided strictly on an "as is" basis and are subject to Section 2 above.

UPGRADES/ADD-ONS

If the Software is designated by PMDG as an Upgrade or Add-On product, you may only use the Software if you are also currently a licensed user of the base product to which the Upgrade or Add-On applies. Unless the PMDG documentation for an Upgrade or Add-On specifically provides, you shall not separate upgrade products from base products, nor transfer them separately. PMDG reserves the sole and exclusive right to set its policies and prices regarding updates, upgrades add-ons and enhancements. All other terms of this EULA apply with equal force to any such Upgrades.

GENERAL/MISCELLANEOUS CLAUSES

This EULA shall be governed by and construed under the substantive laws applicable to the United States of America. If any provision of this EULA is held to be unenforceable, the remaining provisions shall remain in full force and effect.

The undersigned agrees that this EULA is entered into at Sparks, Nevada, and further agrees that any legal or equitable dispute brought or arising under this Agreement shall be brought in the State and Federal courts for the State of Nevada.

This Agreement constitutes the entire Agreement between the undersigned and PMDG about the Software and documentation, and shall not be modified except in a writing signed by PMDG.