

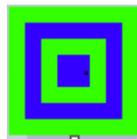
# VMRGenerator

## Manual

Version **6.3**

June 26, 2017

Nico Kaan



Copyright © 2014-2017, Nico W. Kaan, Delft, Netherlands, All Rights reserved.

No part of this publication may be reproduced and/or published by print, photo print, microfilm or any other means without the previous written consent of Nico Kaan.

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Technical information .....	3
1.2	System Requirements .....	3
1.3	Forum .....	3
<b>2</b>	<b>How to install and use VMRGenerator.....</b>	<b>4</b>
2.1	Download and install VMRGenerator.....	4
2.1.1	Running VMRGenerator networked.....	4
2.1.2	Version Check .....	4
2.2	Run VMRGenerator .....	4
2.2.1	Generate matching rules [Optional] .....	7
<b>3</b>	<b>FAQ.....</b>	<b>8</b>
3.1	The program cannot start because MSVCP140.dll and/or VCRUNTIME140.dll is missing from my computer? .....	8
3.2	I'm still getting vPilot red error messages about objects, or even an FSX/P3D crash? .....	8
3.3	How does VMRGenerator create a matching rule? .....	8

# 1 Introduction

VMRGenerator generates Matching Rules for vPilot for all AI Aircraft installed in any FSX or P3D Flight Simulator.

Its main advantages are:

- You will get error free matching rules; it will never happen that vPilot matches a plane you do not have installed!
- If you install or remove (part of) an AI Aircraft package you only have to re-run VMRGenerator.
- When you are using more than one AI package, you may specify what package has priority. Equivalent Airline/TypeCode liveries from lower priority packages will not be put in the rules file.
- Configuring your rules in vPilot is very easy.
- The intermediate file AI\_liveries.txt can be used by users of my PSXseeconTraffic program.

## 1.1 Technical information

VMRGenerator is a 32 bit Windows application, written in C++, for Windows 7 and higher.

## 1.2 System Requirements

In order to run it, you need Microsoft Visual C++ Redistributable for Visual Studio **2015**. The standalone installers for x86 and x64 can be [downloaded here at Microsoft](#). Install the **x86 one**.

## 1.3 Forum

For questions & answers and discussions use my **PSXseeconTraffic** Forum at [MyCockpit.org](http://MyCockpit.org) (free registration required).

## 2 How to install and use VMRGenerator

### 2.1 Download and install VMRGenerator

Unzip **VMRGenerator\_vx.y.zip** where you want at your Flightsim PC. If you upgrade, just overwrite the existing files.

After unzipping, you will find a VMRGenerator folder containing four files. VMRGenerator.exe is the application. Documentation is provided by means of the VMRGenerator\_Manual and the ReleaseNotes.

	<p>An important file in the archive is the EULA.pdf. This End-User License Agreement is a legally enforceable contract between you and the Author. <b>By downloading and installing the VMRGenerator package, you agree to be bound by the terms of this EULA.</b></p>
---	--

*VMRGenerator does not change anything in your FSX/P3D installation. It only reads aircraft.cfg and/or sim.cfg files and it writes into its own folder and/or in the vPilot Files\Model Matching Matching Rule Sets folder at the same PC*

#### 2.1.1 Running VMRGenerator networked

You may also run VMRGenerator from another PC then your FlightSim PC. If you do run networked, use **\\PCname\Sharename** as syntax in your full paths, e.g.

\\ADAM\Prepar3D v3\Simobjects\WoAI

Where Prepar3D v3 is a sharename on my flightsim PC with name ADAM

#### 2.1.2 Version Check

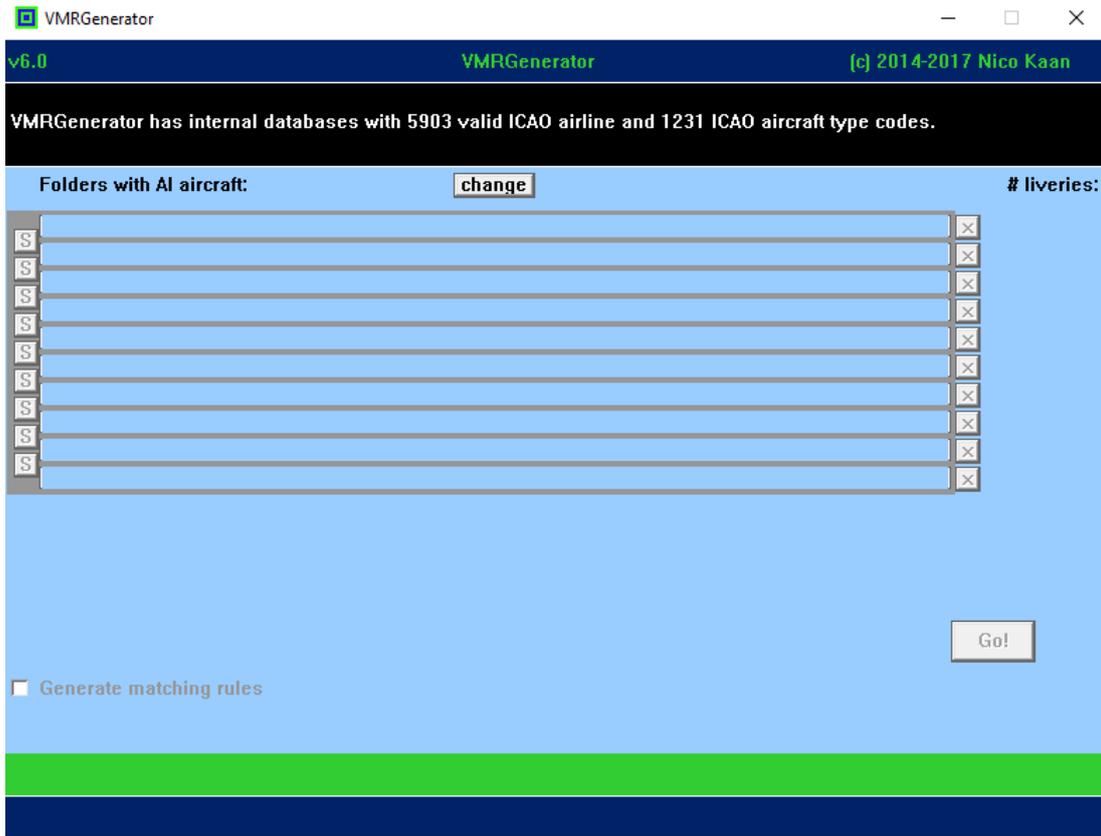
At each start-up VMRGenerator will check if there is a new version available. If so the – normally- green version message in the main window will turn yellow, and the new version number will be indicated. This is just a reminder, you have to download and install the new version yourself.

If you do not want this functionality, you can switch it off by adding this line to VMRGenerator.cfg (in the same folder as VMRGenerator.exe):

**VERSION\_CHECK=0**

### 2.2 Run VMRGenerator

At start-up the main window will show like this:



VMRGenerator has an internal databases with **5903** valid ICAO airline codes and **2418** valid ICAO aircraft type codes.

If you find that a code is missing you can post that in my [PSXseeconTraffic forum](https://www.psx-secon.com/forum/). Then I will add these in the next version.

With the **change button** you can enable the folders area for editing:

- Add a folder with AI aircraft by typing the full-path name into a free folder edit text box, e.g.

**E:\Program Files (x86)\Lockheed Martin\Prepar3D v3\SimObjects\WoAI**

When you are finished editing, push the change button again in order to make the change effective. VMRGenerator will check immediately if the folder exists.

- You can re-order the folders contents with the **Swap buttons**.

*The higher the position of the folder, the higher its priority. If an airline-aircraft type is available in two or more folders, only the highest priority one will be put in the AI\_liveries.txt and vPilot rules files.*

- You can delete a folder with the **X button**.

If all path names exist, the **Go!** button in the right bottom corner will be enabled. The paths to your AI folders will be saved in VMRGenerator.cfg.

Push the Go! button, if you're satisfied with the folders!

VMRGenerator then searches for **aircraft.cfg** or **sim.cfg** files in the folders specified. It reports for each folder how many valid liveries were found.

After all folders have been searched it will write all liveries to an intermediate file called **AI\_liveries.txt**. While doing that it ignores Aircraft/TypeCode liveries that are already read from a higher priority folder. Liveries that were ignored will be written to the file **Liveries\_ignored\_by\_priority.txt**.

*The intermediate file AI\_liveries.txt is a plain text file. Each line contains a livery found. First the Airline code, then the TypeCode, then the folder in which it was found (1 is the first folder in the folders area), then the title of the livery and finally (optionally) the registration code(-s). The items are separated by semicolons.*

If there are invalid Airline codes or invalid Aircraft Type Codes found, VMRGenerator writes these into two files in the VMRGenerator folder: **InvalidAirlineCodes.txt** and **InvalidAircraftCodes.txt**. With the information in those files it will be easier for you to find, and manually repair these errors in your AI packages, if you want to make the best out of your AI package, otherwise you can just go on without paying attention to it.

*Note: To prevent warnings for Invalid Arline Codes for non-airline aircraft, such as General Aviation liveries, write: **atc\_parking\_codes=PVT** for that livery in aircraft.cfg*

VMRGenerator also checks if the **texture=**, **model=** and **sim=** values in a [fltsim.x] section exist, if not the livery is skipped; this prevents completely black aircraft and/or can't create messages.

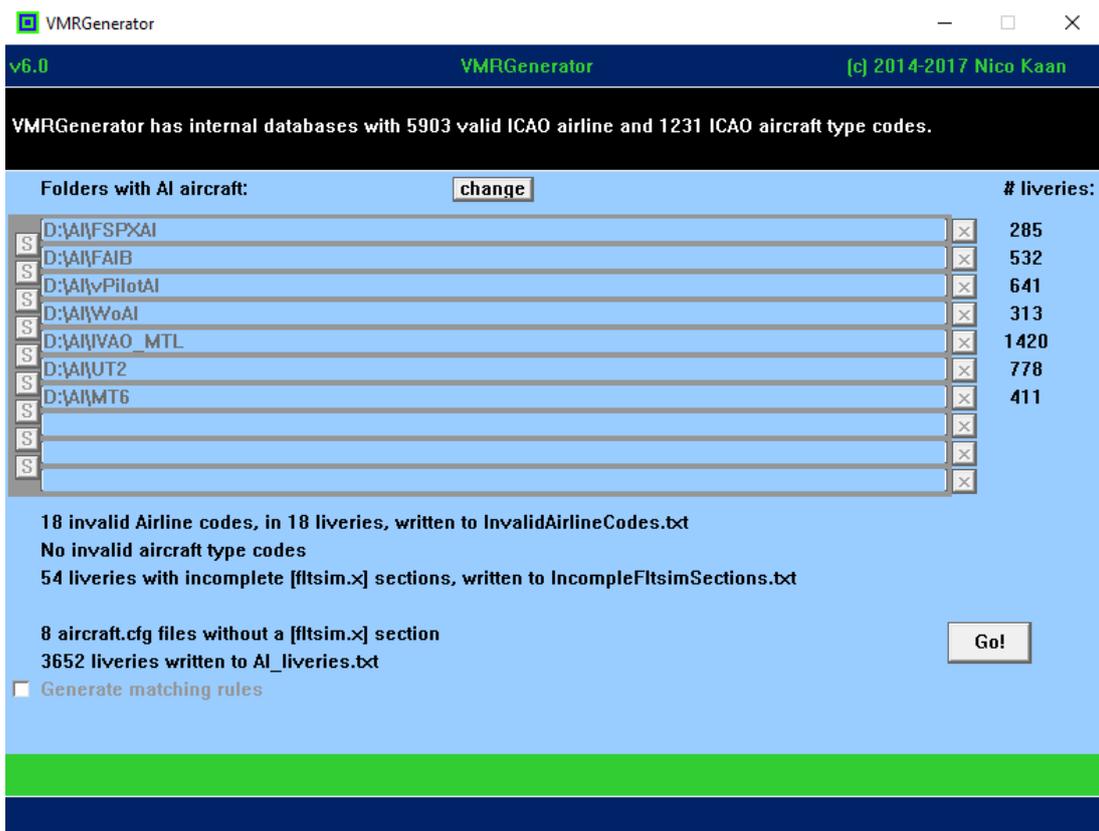
A livery is also skipped if there is no value for the **atc\_parking\_codes=** parameter or for IVAO\_MTL if there is no Airlinecode in the value for the **title=**. In the main interface a message will give the number of liveries skipped because of incomplete fltsim sections and the skipped liveries are written to the file **IncompleteFltsimSections.txt**.

A complete aircraft.cfg file is ignored if their [flysim.x] sections are not numbered consecutively from 0 to n and/or if there are no [fltsim.x] sections in it.

VMRGenerator also detects the **Registration** code(-s) of an aircraft. It takes the value of the **atc\_id=** line in aircraft.cfg or sim.cfg as aircraft registration code. On this [website](#) you will find the registration codes for lots of airlines.

*Normally a livery has of course only one unique Registration code. However, you may fill in multiple Registration codes separated by comma's. Thus it is easier to separate different groups of liveries for a single type without having to install all liveries. For example to differ between Normal BAW 744 liveries and BAW Oneworld 744 liveries. Or to distinguish all freighter liveries of a certain type from the Pax liveries of that same type, or 773-ER versions from 773 versions.*

Searching Liveries will not take long and you will see a screen like this:



As you can see IVAO\_MTL is supported for generating liveries because VMRGenerator is also used by IVAO PSXseeconTraffic users. However, vPilot users may not use it. Matching rule generation is blocked if you are using IVAO\_MTL liveries (read more [here](#)).

The **VMRGenerator.log** file gives a more detailed overview of the run.

### 2.2.1 Generate matching rules [Optional]

*By default the rule generation phase is not activated. This is because there are many more people that use VMRGenerator for PSXseeconTraffic than for vPilot.*

The Rule generation phase starts, if the Rule Generation checkbox is selected, with VMRGenerator reading the AI\_liveries.txt file.

The resulting rule file **AI Matching Rules Generated.vmr** is written into the VMRGenerator folder. If VMRGenerator detects a vPilot rules folder it will automatically put a copy into that folder too.

Add the rules file **AI Matching Rules Generated.vmr**, to the Downloaded Rule Sets of vPilot as follows: Run vPilot, select Settings, Model Matching tab and add the file by clicking the Add Custom File(s)... button and select the file from the 'vPilot Files\Model Matching Rule Sets folder'.

Do not forget to use a title from one of the aircraft in AI Matching Rules Generated as Default title, and you may delete other rules in your Downloaded Rule Sets.

Note that you only have to do this Custom File addition once. If you run the VMRGenerator again, you do not have to anything with the vPilot Model Matching settings!

## 3 FAQ

### 3.1 The program cannot start because MSVCP140.dll and/or VCRUNTIME140.dll is missing from my computer?

See section 1.2 System Requirements.

### 3.2 I'm still getting vPilot red error messages about objects, or even an FSX/P3D crash?

Possible causes:

- If you get lots of them: You probably have forgotten to activate the AI package in your Flight Simulator. For P3Dv3 for instance you should add a path statement for each package in c:\Program Data\Lockheed Martin\Prepare3D v3\simobjects.cfg.
- Although the aircraft is installed in FSX/P3D, creating an object failed (due to incorrect data), or even worse, caused FSX/P3D to crash...
- You may have deleted some [fltsim.x] sections from aircraft.cfg files in AI Packages but you forgot to renumber these [fltsim.x] sections starting from 0, consecutively... Note that there is a nice auto-renumbering tool, named FSReNum, for that. Just put it in the folder of the aircraft.cfg file and run it once.

### 3.3 How does VMRGenerator create a matching rule?

VMRGenerator deduces a rule based on information it finds in the aircraft.cfg or sim.cfg file. It searches for the lines with:

parameter	is used as
atc_model	ICAO aircraft code or TypeCode in vPilot terminology
title	Title of livery or ModelName in vPilot terminology
atc_parking_codes	ICAO Airline code(-s) or CallSignPrefix in vPilot terminology

The IVAO package does not support all these parameters, but thanks to their rigid information structure VMRGenerator can generate rules for their aircraft. However, you must have installed IVAO in its default folder: ..\SimObjects\IVAO\_MTL or ..\IVAO\_MTL, in order to be recognised by VMRGenerator.

For IVAO liveries are recognised based on:

parameter	is used as
max 4 letters after "IVAO_" in the folder name	ICAO aircraft type code
atc_id	Registration code(-s)
title	Title of livery
first 3 letters after the aircraft type code in the title	ICAO airline code