Identifier:	Revision:	Effective Date:
001.001	**3**	**02/09/2016**

Document Catalog Number:	**n.a.**

Author: geniX soft

FSXAISConnector Technical Specification

geniX Soft

Visit at www.genixsoft.com

Document History

Revision History

Revision Number	Revision Date	Summary of Changes	Author
0	08/02/2011	First release	geniX
1	08/20/2011	First revision	geniX
2	02/08/2016	Second revision	geniX
3	02/09/2016		geniX

Approvals

This document requires following approvals:

Name	Title

Distribution

This document has been distributed to:

Name	Title

Table of Contents

1. Introduction

1.1.　Purpose

The FSXAISConnector (**Connector**) is a generic name that would indicate a Win32 program that is able to generate a couple of Comma Separated Value (CSV) files (**traffic files**) for using by FSX AI Spotter (**FSXAIS**) application developed by geniX software for integrated in Microsoft Flight Simulator FSX®.

The content of these CSV files could be scheduling information of airlines at a certain airport, for both departures and arrivals operation, in a manner that FSXAIS is able to display the appropriate aircraft in the more appropriate airport in the Microsoft Flight Simulator FSX® (**MS-FSX**).

Morevoer, the developing of Connector interfaces from Web Services Providers of airline flight informations, will bring the possibility to have on-line true current airport traffic and, using FSXAIS, display this *live* world in the MS-FSX.

We suggest for a Win32 Console Application model for the connector.

1.2.　Scope

Scope of this document is to illustrate the CSV required schema and the input parameters of the Connector.

2. Describing the System

2.1.　Key Features

The Connector must be a Win32 Console Application. This application will be recalled from FSXAIS at regular interval (min each 15 minutes, configured from FSXAIS GUI) in a Console shell and must provide as output the traffic files as described further in this document.

Anyway, FSXAIS provide an "offline" modal in which no external connector is executed, but it simply read the text files provided in the main application directory[1]. In this mode, every user is able to create their own application that can provide the traffic files for FSXAIS.

All the collected information are to be stored in a CSV file format.

2.2.　Architecture overview

Here is a general architecture overview:

---
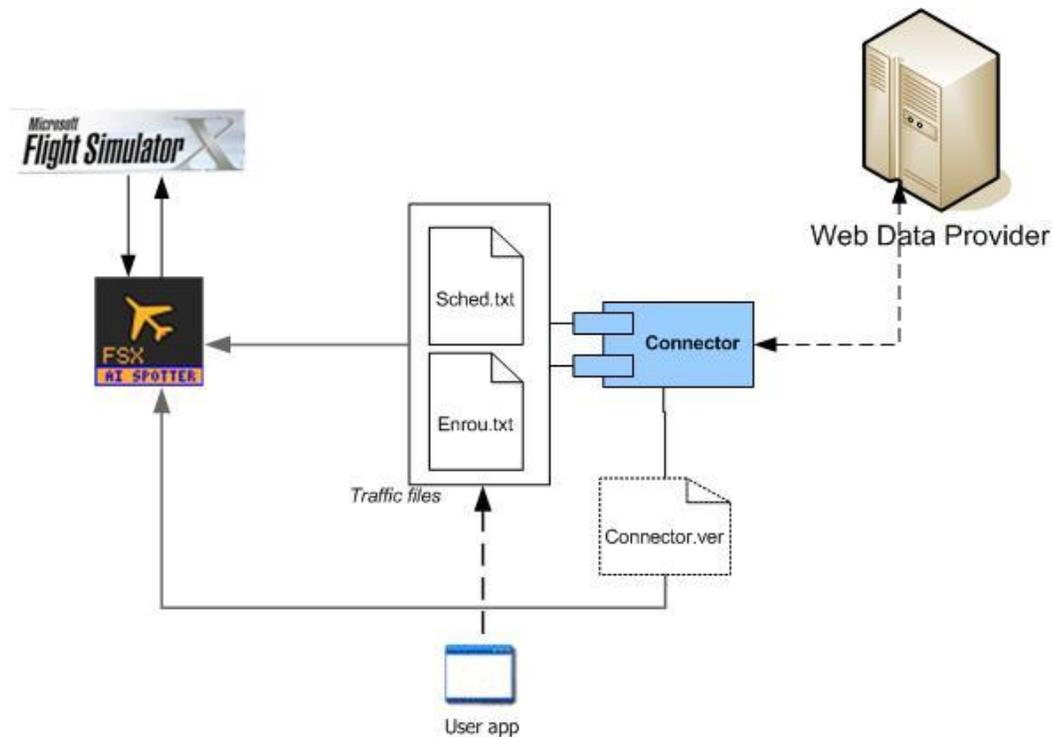
[1] Using the same timer interval

In this schema, the Connector can connect to the web and provides 2 traffic files:

- Sched.txt, containing the list of departures flights

- Enrou.txt, containing the list of arriving flight

A third files is used for describe the connector itself (used by FSXAIS GUI):

- Connector.ver, a text file containing application informations

As illustrated before, when FSXAIS is in offline mode, a general user application could provide traffic files at regular interval. It is in charge of FSXAIS the loading of the traffic files and their management.

3. Connector procedures and integration with FSXAIS

3.1.   The connector

The Connector application accept **some parameters** on the command line. The forms is:

Fsxconnector fnct-idx [user pwd] [icao] [rows]

Where:

Fnct-idx = function index to call for the connector (see next)

User = user name for access the web service provider

Pwd = password for the user name above

Icao = airport icao code to query for info on web service provider

Rows = max number of items to return in the .txt files

Obviously, the FSXAIS application will only call the Connector program using the above provided parameters.

3.2.   Installing the connector

It is not mandatory to have a full setup for connector installation.

The prerequisite is that the connector must resides in the same directory of FSXAIS installations as long as the output files generated.

3.3.   Starting the connector

FSXAIS has a timer functionality for calling the connector executable with the appropriate parameter.

For GUI based connectors, FSXAIS allow an off-line mode where it simply try to load the traffic files instead of make an external shell command to run the connector executable; this at regular interval defined by the user, anyway.

4. Connector functions

4.1.   Main Functions (fnct-idx)

Basically, the connector must provide the following basic functions:

Function "**0**": connector should generate the file connector.var containing *Class, description* and *files names* (traffic files) of the connector itself (see next for further info).

Function "**1**": connector should generate 2 txt files: sched.txt and enrou.txt with flights informations.

Examples:

Connector **1** jmack ints6 KLAX 25

will generate **both** the files with flights departing and arriving at Los Angeles Airport (KLAX), with a maximum of 25 rows in each file. "jmack" and "ints6" are supposed user and pwd to access the Web Service to get the live information.

Connector **0**

will generate the text-data file connector.ver

4.2.   Errors and malfunctions

Actually, is the FSXAIS application that is in charge to check for errors and malfunctions of the connector , like timeout and empty txt files.

However, in the FSXAIS GUI is possible to configure a timeout for the connector. When its raise the tresholds, FSXAIS will emit a warning window.

5. Connector files templates

It is responsibility of the connector developer to ensure that the files created are suitable for FSXAIS application for both informals and operationals aspects. The example below are for the Los Angeles Airport (KLAX).

5.1.   Departure file (*"sched.txt"*)

This CSV file is structured as follow:

ICAO Code+Flight number, ICAO Aircraft Type, Dep. Apt, Arr. Apt, Date UTC Time, *Aircraft Reg.#*

Example:

SKW469,E120,KLAX,KDFW,30/07/2011 17.33.00, *N23456*

where:

SKW469 is the ICAO airline code+flight

E120 is the aircraft type (ICAO defined)

KLAX is the ICAO code for the departure airport (as we call the connector for flight from KLAX, as example)

KDFW is the ICAO code for the destination airport

30/07/2011 17.33.00 is the date and UTC time of the departure

N23456, is the **optional** tail of the aircraft

NOTE: The Time string separator may be "." (as in the example before) or ":"; the date separator is based on "locale".

5.2. Arrival file (*"enrou.txt"*)

This CSV file is structured as follow:

ICAO Code+Flight number, ICAO Aircraft Type, Dep. Apt, Arr. Apt, Date UTC Time, *Aircraft Reg.#*

Example:

DAL1934,B752,KJFK,KLAX,30/07/2011 18.29.00, *N65432*

where:

DAL1934 is the ICAO airline code+flight

B752 is the aircraft type

KJFK is the ICAO code for the departure airport

KLAX is the ICAO code for the destination airport (as we call the connector for flight from KLAX, as example)

30/07/2011 18.29.00 is the date and UTC time of the departure

N65432, is the **optional** tail of the aircraft

NOTE: The Time string separator may be "." (as in the example before) or ":"; the date separator is based on "locale".

5.3. connector.ver file

This data-text file is structured as a typical **.ini file**. It contains two rows as follow:

CLASS=C2

DESCR=geniX Software - Connector for FlightAware FlightXML API rel.1.1

DEP=sched.txt

ARR=enrou.txt

where:

CLASS is the key that establish the particular "class" to which belongs the connector. This value must be in the form of **Cx**, with x the class-value defined for the connector (see next).

DESCR is the key that contain the connector description.

DEP is the key for the name of the file that will be generate for the departure.

ARR is the key for the name of the file that will be generate for the arrivals.

The above values are used by the FSXAIS application.

6. FSXAIS and the traffic files

From the FSXAIS perspective, the traffic files has a key composed by the airline ICAO code (3 letters) plus the flight number. Each time the traffic files are loaded in a FSXAIS session, the datas are matched using just the key composed as above.

If present, the aircraft tail value is used to match an AI aircraft that is present in the user' FSX *Simobject* directory, otherwise, the rule is to use the first (AI) aircraft in the FSX that:

- Has the ICAO code (3 letters) = atc_parking_codes in the aircraft.cfg

- Has the ICAO aircraft type = atc_model in the aircraft.cfg

See FSXAIS User guide for more information.

The most important things is that FSXAIS assumes:

1. **Departure traffic file contains <u>only</u> aircraft that are already on the ground (at indicated airport gate).**

2. **Arrival traffic file contains <u>only</u> aircraft that are in the air having the indicated airport as destination.**

7. Appendix A – Class of connectors

FSXAIS must be aware of the capabilities of the connector program that is going to call.

For this reason, we have stated **2 classes** for the connector program, based on 4 operational capabilities.

CLASS ID.	Departures func.	Arrivals func.	Tail availab.
1	**YES**	**YES**	YES
2	**YES**	**YES**	NO

Departures and Arrivals functionaly are obviously mandatories.

So, for example, a connector program tha provides only Departure and arrivals functionality, is to be defined as *Class 2 Connector*.