FMS Data Manager Automagic Scanner Rules

The FMS Manager handles Flight Simulator add-ons (most of the aircrafts users can fly in Microsoft Flight Simulators), X-Plane add-ons (most of the flyable aircraft in X-Planes), and tools (the software that consumes our navdata but does not appear in flight simulators) differently. Thus we have the rule below:

Microsoft Flight Simulator (MSFS) addons

The addon search for MSFS addons are divided into 2 parts: search for simulator path, search for aircraft.cfg file. Thus the search section in index.xml should always contains <directory> and <pattern> section The <directory> section tells in which directory should the data install into, and the pattern should be the regular expression that appeared in Aircraft.cfg. For instance, to search for Captain Sim's product can be achieved by the code here: http://codebeautify.org/xmlviewer/cb0348c5

X-Plane addons

As the X-Plane does not provide a readable aircraft.cfg file, and the aircraft addons can be placed anywhere, the FMS Data Manager attempt to find the addons by looking for a specified file that belongs to that addon which assigned in <filesystem> section. Since the navdata directory may be different from the main program, <directory> section is used for the data installation. The search for addon JARDesign A320neo could be considered as an example: http://codebeautify.org/xmlviewer/cb3e4ca4

Tools

The search for tools are different, as the data folder might not sit along with the flight simulators. Thus the FMS Data Manager might look into different directories by the conditions provided in <search> section. It is the most complicated search, that may involve multiple directories and/or registry paths. If it is required to identify if a file exists in a particular directory, it uses <filesystem> section; If it's resides in Windows Registry, <registry> section should be filled. Here is an example for PFPX: http://codebeautify.org/xmlviewer/cb5ccbb1

Placeholders that mostly used

Simulator name	**Simulator argument in data.index, e.g. simulator="XP9"**	**Data installation directory: Path placeholder in data.index <directory> e.g. path=”[SIMROOT]\ClassicJetSimUtils"**	**Search for tool (e.g. ActiveSky, Pro ATC, etc.) installations: Path placeholder in data.index <filesystem> e.g. path="[XP9ROOT]\Aircraft\”**	**[simulators] in settings.conf**
Flight Simulator 9	FS9	[SIMROOT]	[FS9ROOT]	FS9
Flight Simulator X	FSX	[SIMROOT]	[FSXROOT]	FSX
Prepar3D	P3D	[SIMROOT]	[P3DROOT]	P3D
Prepar3D v2	P32	[SIMROOT]	[P32ROOT]	P32
Prepar3D v3	P33	[SIMROOT]	[P33ROOT]	P33
X-Plane 10	XP10	[SIMROOT]	[XP10ROOT]	XP10
X-Plane 11	XP11	[SIMROOT]	[XP11ROOT]	XP11
X-Plane 9	XP9	[SIMROOT]	[XP9ROOT]	XP9
Flight Simulator X Steam Edition	FSXS	[SIMROOT]	[FSXSROOT]	FSXS

Additional placeholders for tool search

More information here: http://doc.qt.io/qt-5/qstandardpaths.html

Placeholder	Description
[DOCUMENTS]	Returns the directory containing user document files.
[APPDATA]	Returns the directory containing the user applications.
[APPDATASHARED]	Returns a directory location where persistent data shared across applications can be stored.
[APPCACHE]	Returns a directory location where user-specific non-essential (cached) data should be written.
[APPCACHESHARED]	Returns a directory location where user-specific non-essential (cached) data, shared across applications, should be written.
[APPCONFIG]	Returns a directory location where user-specific configuration files should be written.
[APPCONFIGSHARED]	Returns a directory location where user-specific configuration files shared between multiple applications should be written.
[HOME]	Returns the user's home directory.
[DESKTOP]	Returns the user's desktop directory.
[DOWNLOADS]	Returns a directory for user's downloaded files.
[RUNTIME]	Returns a directory location where runtime communication files should be written.
[TEMP]	Returns a directory where temporary files can be stored.

[SYSTEMDRIVE]	Returns the system drive where operation system is installed.
[PUBLIC]	Returns the public data location, for example C:/Users/Public/
[PROGRAMDATA]	Returns the program data location (Windows only) for example C:/ProgramData/
[COMMONPROGRAMFILES(X86)]	Returns the program folder location (Windows only), for example C:/Program Files (x86)/
[NONE]	Returns empty string

XSD definition for index.xml:

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="addonIndexes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="addon" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="mapping" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="directory">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:string" name="path" use="optional"/>
                            <xs:attribute type="xs:string" name="fromregpath" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
```

```
<xs:element name="search" minOccurs="0">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded" minOccurs="0">
      <xs:element name="pattern">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="value" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="uninstall">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="guid" use="optional"/>
              <xs:attribute type="xs:string" name="key" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="filesystem">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="path" use="optional"/>
              <xs:attribute type="xs:string" name="filename" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
      <xs:element name="registry">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute type="xs:string" name="path" use="optional"/>
              <xs:attribute type="xs:string" name="value" use="optional"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
```

```
                    </xs:element>
                  </xs:choice>
                </xs:complexType>
              </xs:element>
              <xs:element name="config" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="entry">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension base="xs:string">
                            <xs:attribute type="xs:string" name="key" use="optional"/>
                            <xs:attribute type="xs:byte" name="value" use="optional"/>
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute type="xs:string" name="path" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="platform" use="optional"/>
            <xs:attribute type="xs:string" name="simulator" use="optional"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute type="xs:string" name="name" use="optional"/>
      <xs:attribute type="xs:string" name="guid" use="optional"/>
      <xs:attribute type="xs:short" name="cycle" use="optional"/>
      <xs:attribute type="xs:byte" name="revision" use="optional"/>
      <xs:attribute type="xs:string" name="masterfile" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute type="xs:string" name="id"/>
<xs:attribute type="xs:short" name="cycle"/>
<xs:attribute type="xs:string" name="internalId"/>
<xs:attribute type="xs:string" name="fileRevision"/>
</xs:complexType>
</xs:element>
```

```
</xs:schema>
```