AeroWeather Exchange Format

Version 1.1 / 2.10.2017

This document defines a file exchange format for AeroWeather:

1. Backup and restore the user data
2. Exchange of user data between various aero applications (in future releases)
3. Exchange of user data between third-party applications (1)
4. Provisioning of custom data sets (e.g., airlines)
5. Giving the user the ability to export, mutate and re-import data. E.g. to create notes of stations on desktop computer and then import them into AeroWeather.

What data are included:

Groups, Stations, Notes, (Preferences)

General conditions:

- JSON Format (2)
- All Inputs are Strings (except when noted)
- Filename extension ".aw2"

(1)   Eg. AeroWeather for Android from www.sparklingapps.com
(2)   https://en.wikipedia.org/wiki/JSON

Basic Structure

```
{
      "metadata": {
            …
      },
      "groups": [{
            …
      }],
      "notes": [{
            …
      }]
}
```

Basic structure of the aero file exchange format in JSON

Metadata

Example of the metadata structure and their definitions:

```
"metadata": {
        "contentType": "backup",
        "creatorBundleID": "com.lakehorn.AeroWeatherNG",
        "creatorBuild": "19",
        "creatorName": "AeroWeather Pro",
        "creatorDevice": "iPad",
        "createdDate": "2017-07-18T15:54:47GMT+02:00",
        "creatorLanguage": "en",
        "creatorPlatformVersion": "10.3.2",
        "version": "1",
        "creatorPlatform": "ios"
    },
```

contentType *required*:

This describes how the data was produced. Possible values are:
- backup: file contains all user data of the app
- exportLite: file contains all stations (always generated from AeroWeather Lite)
- Group: file contains one or more groups
- snapshot: for internal use only

creatorBundleID *required*:

A unique App identifier over all platforms. On Apple iOS and macos systems the bundle ID is used.

creatorBuild:

The build version of the source which created the file, e.g.,"19" (letter are possible too).

creatorName *required*:

The name of the program which created the file, e.g., "AeroWeather Pro"

creatorDevice:

The name of the device type, e.g., "iPhone", "iPad", "Galaxy"

createdDate:

The date (timestamp) when the file is created. The required format is defined in
https://en.wikipedia.org/wiki/ISO_8601
ISO8601 DateFormat = "yyyy-MM-dd'T'HH:mm:ss'Z'"
Example: 2017-07-18T13:49:58GMT+02:00

creatorLanguage:

The app language that the user specified. Locale tags will be IETF language tags.
https://en.wikipedia.org/wiki/IETF_language_tag

creatorPlatformVersion:

The version of the operating system when the data was generated, e.g., "10.3.2"

version *required*:

The version of the data exchange format. E.g. if exchanged data expanded or definitions are changed the version number increases. As changes to the data exchange format are described in this document, the version number of the data exchange format corresponds to the version number to this document.

creatorPlatform:

One of the following strings: "android", "macos", "ios", "tvos", "watchos"

Groups

Example of the group's structure and their definitions:

```
"groups": [{
        "name": "ZRH-NRT",
        "position": 0,
        "sortOrder": "userDefined",
        "dataFormat": "decoded",
        "locationCodeType": "locCode",
        "style": "regular",
        "locations": ["CHLSZB", "JPRJAA", "USKLAS"],
        "identifier": null
    }]
```

name *required*:

Name of the group.

position:

Position (Integer) of group within root group

sortOrder:

Defines how the locations are ordered. Possible values are: "userDefined", "name", "icaoCode", "iataCode", "countryWithName", "countryWithIcao", "proximity"

dataFormat:

Defines how METARs are shown in the group list. Possible values are: "raw" or "decoded"

locationCodeType *required*:

Defines the code type of the location listed in the locations (see below) attribute. Possible values are: "ICAO", "locCode". The type "locCode" uses an internal code. Usually the type "ICAO" should be used.

style:

Defines the style of METAR in the group list. Possible values are: "regular", "timeBased"

locations (Array) *required*:

Array of location codes as defined in locationTypeCode.

identifier:

Unique identifier of a group. Is not required.

Notes

Example of the notes structure and their definitions:

```
"notes": [{
        "locationCodeType": "locCode",
        "location": "CHLSZH",
        "title": "Zurich",
        "text": "GNSS approaches are possible now.",
        "createdDate": "2017-02-13T15:14:58GMT+01:00",
        "color": "orange"
    }]
```

locationCodeType *required*:

Defines the code of the location listed. Possible values are: "ICAO", "locCode". The type "locCode" uses an internal code. Usually the type "ICAO" should be used.

location *required*:

Code of the location as defined in locationTypeCode.

title :

Title of the note.

text *required*:

The content of the note.

createdDate:

The date (timestamp) when the note was created. The required format is defined in
https://en.wikipedia.org/wiki/ISO_8601
ISO8601 DateFormat = "yyyy-MM-dd'T'HH:mm:ss'Z'"
Example: 2017-07-18T13:49:58GMT+02:00

color:

Color category of the note. Possible values are "red", "orange", "green", "blue", "purple", "grey", "black"

Examples:

Example for contentType "group" to transfer a group

```
{
    "metadata": {
        "contentType": "group",
        "creatorBundleID": "com.lakehorn.AeroWeatherNG",
        "creatorBuild": "19",
        "creatorName": "AeroWeather Pro",
        "creatorDevice": "iPad",
        "createdDate": "2017-07-18T15:55:26GMT+02:00",
        "creatorLanguage": "en",
        "creatorPlatformVersion": "10.3.2",
        "version": "1",
        "creatorPlatform": "ios"
    },
    "groups": [{
        "name": "ZRH-NRT",
        "sortOrder": "userDefined",
        "dataFormat": "decoded",
        "locationCodeType": "locCode",
        "style": "regular",
        "locations": ["CHLSZB", "JPRJAA", "USKLAS"],
        "identifier": null
    }]
}
```

Example for contentType "backup"

```
{
      "metadata": {
            "contentType": "backup",
            "creatorBundleID": "com.lakehorn.AeroWeatherNG",
            "creatorBuild": "19",
            "creatorName": "AeroWeather Pro",
            "creatorDevice": "iPad",
            "createdDate": "2017-07-18T15:54:47GMT+02:00",
            "creatorLanguage": "en",
            "creatorPlatformVersion": "10.3.2",
            "version": "1",
            "creatorPlatform": "ios"
      },
      "groups": [{
            "name": "Ohio 🇺🇸airfields",
            "position": 0,
            "sortOrder": "countryWithName",
            "dataFormat": "decoded",
            "locationCodeType": "locCode",
            "style": "regular",
            "locations": ["USKFAY", "USKSFO", "USKCMH", "USKFBL", "USKUNI",
"USKFAT", "USKCAK", "USKCVG"],
            "identifier": "B9F6B7B6-E210-4B34-967B-379E7F9DB376"
      },{
            "name": "Brasil",
            "position": 1,
            "sortOrder": "userDefined",
            "dataFormat": "decoded",
            "locationCodeType": "locCode",
            "style": "regular",
            "locations": ["BRSBJD" , "BRSBGR"],
            "identifier": "2B754328-6726-453C-8842-32159105CF32"
      }],
      "notes": [{
            "locationCodeType": "locCode",
            "location": "CHLSZH",
            "title": "Zurich",
            "text": "GNSS approaches are possible now.",
            "createdDate": "2017-02-13T15:14:58GMT+01:00",
            "color": "orange"
      }]
}
```

Document Versioning

19-Jul-2017:   Initial Version
02-Oct-2017:   Added position of group within list in NGDocGroup

License

The AeroWeather Exchange Format (AWEF) is made available as open format which is platform independent, machine readable, and made available to anybody without restrictions.