

LoggerDownloader

Marc André | marc.andre@netline.ch | Skype: mandre.bern

LoggerDownloader Version 1.1.7

Generic

The logger downloader is very flexible. It tries to identify all relevant track files in a folder and can act on them like copy, move, etc.

Identification of the relevant track files is done based on date and time. All files that contain track points within given time interval are selected.

In addition the pilot is identified based on serial number of the logger. This process is optional.

Installation

LoggerDownloader requires .NET Framework 3.5 be installed. This should be present on Windows 7 and later. Previous systems can download it from Microsoft.

Copy all files into a folder and run LoggerDownloaderGUI.exe

License

The logger downloader may freely be used. If you are interested to participate in its development, I am happy to share source code.

Configuration (config.xml)

Configuration of the software is done with config.xml. This file must be present in the same folder as the LoggerDownloaderGUI.exe. The configuration file contains three sections.

- EventConfig is used to configure event specific settings such as the event code and the morning/evening flight time intervals
- InputProcessors is used to define the input of tracks. At this time only FAI/CIA Balloon Loggers are supported
- OutputProcessors are used to configure the actions on the identified track files.

Please note that the configuration is only loaded at program start!

EventConfig

```
<EventConfig Code="" PilotsFile="X:\Misc\Software\LoggerDownload\Pilots.xml">
  <FlightTimes>
    <FlightTime Name="AM" Start="05:00" End="12:00" />
    <FlightTime Name="PM" Start="15:50" End="22:00" />
  </FlightTimes>
</EventConfig>
```

- EventConfig Code can be used for file naming, see variable below.
- PilotsFile is used to identify the logger based on Serial Number. See description below. Set empty or remove tag to omit this step.

- Flight times are time intervals for given flight slots. The flight slots will be presented to the user in a drop down. Usually AM and PM are defined. Times are in local time.

InputProcessors

```
<InputProcessor>
  <!-- Input processor used for reading the logs -->
  <BalloonLoggerProcessor SourcePath="h:\" UTCOffset="7200"/>
</InputProcessor>
```

At this time only Balloon Competition Loggers are supported. Required Parameters are:

- SourcePath which is usually the drive of the SD card
- UTCOffset is the Offset to UTC in seconds (as defined in the logger.cfg)

OutputProcessors

A number of output processors are defined. Output processors act on the identified track files and can copy, move and delete them. In many processors the destination file can be specified using variables. The available variables are specified below.

Variables

%f() ¹⁾	Flight number.
%p() ¹⁾²⁾	Pilot number
%s	Logger Serial number
%i()	Logger Pilot ID (Pilot ID from IGC header)
%e	Event Code
%t	Flight Time Code
%l() ¹⁾	Track number
%o	Original file name without file extension

¹⁾ Variables with parentheses () allow for the specification of the minimum number of digits to be used. If parentheses are omitted, 1 digit is assumed.

²⁾ Only available if pilot list is used

Output Processors

Output processors are called when the input processors successfully identified track files. The output processors are executed in sequence of their definition in the file. The same output processor can be instantiated multiple times with different parameters.

The following output processors are defined:

CopyProcessor

```
<CopyProcessor Destination="d:\Flights%\f%(3)\f%f(3)_p%p(3)_l%1.igc" Override="true"
SetReadOnly="false"/>
```

Copies all identified track files to a folder with a specified filename. The folder is automatically created if missing.

Parameters

Name	Description	Required
Destination	Destination path and filename	Required

	Variables are allowed	
Override	Allow overriding of files? If true and file exists, copy will fail.	Optional, Default=false
SetReadOnly	Set read-only flag to true after copy	Optional, Default=false

MoveProcessor

```
<MoveProcessor Destination="d:\move\%f(3)_p%p(3)_l%l.igc" SetReadOnly="false"/>
```

Moves all identified track files to a folder with a specified filename. The folder is automatically created if missing.

Name	Description	Required
Destination	Destination path and filename Variables are allowed	Required
SetReadOnly	Set read-only flag to true after copy	Optional, Default=false

CopyMainTrackProcessor

```
<CopyMainTrackProcessor Destination="d:\Reckless\%f(3)\f%f(3)_p%p(3).igc"
ValidPoints="1800" Override="true"/>
```

Copies the primary track file for each pilot. This processor can be useful for batch processing. The main track is identified if it contains a minimum given number of valid points. The processor fails if multiple track files match this criteria.

Name	Description	Required
Destination	Destination path and filename Variables are allowed	Required
ValidPoints	Minimum number of valid points that the trackfile must contain.	Optional, Default=false
Override	Allow overriding of files? If true and file exists, copy will fail.	Optional, Default=false
SetReadOnly	Set read-only flag to true after copy	Optional, Default=false

ManualCopyProcessor

```
<ManualCopyProcessor Source="h:\*.*" Destination="d:\backup\SDCards\Flight
%f(3)\Pilot %p(3)\ " Override="false" SetReadOnly="true"/>
```

Copies files based on wildcards (independent on the identified track files) to a destination folder which is named based on variables. A classic example of this command would be to create a backup of the full SD card for each flight.

Name	Description	Required
Source	Source file(s) Variables are not allowed, but wildcards (*) can be used.	Required
Destination	Destination path (without filename) Variables are allowed, except for %l and %o.	Required
Override	Allow overriding of files? If true and file exists, copy will fail.	Optional, Default=false

SetReadOnly	Set read-only flag to true after copy	Optional, Default=false
--------------------	---------------------------------------	-------------------------

ManualDeleteProcessor

```
<ManualDeleteProcessor File="h:\dcache.dat" />
```

Deletes files based on wildcards. A classic example would be the deletion of dcache.dat

Name	Description	Required
File	File name(s) to delete Variables are not allowed, but wildcards (*) can be used.	Required

PilotIDTestProcessor

```
<PilotIDTestOutputProcessor />
```

This processor doesn't actually work on the track files, but only checks if the PilotID as set by the logger matches the pilot number in the pilots list for all identified track files.

This processor doesn't have any input parameters. It is recommended to run this processor as first and only if the track file's pilot ID is used.

Pilots (pilots.xml)

The pilots.xml (or named as defined in the EventConfig section) is used to provide the logger assignment to the Logger Downloader Utility.

Please note that the pilots list is only loaded at program start. Using the pilots list is optional.

```
<?xml version="1.0" encoding="utf-8"?>
<Pilots>
  <Pilot Name="Neil Gabriel" Number="7" LoggerSerial="5AI" />
  <Pilot Name="Henk Broeders" Number="1" LoggerSerial="5AC" />
  <Pilot Name="Stefan Zeberli" Number="3" LoggerSerial="5AE" />
</Pilots>
```

Each Pilot tag contains the following parameters

Name	Description	Required
Name	Pilot Name as shown on the GUI	Required
Number	Pilot Number. This number will be used as %p variable	Required
LoggerSerial	Serial number of the logger used to identify the pilot	Required

Xml is very sensitive on correct file format. Ensure that no special characters are used in the names or that the file is correctly saved in UTF-8 format.