

AMENDMENT LIST NUMBER 5 TO
TECHNICAL SPECIFICATION
FOR IGC-APPROVED GNSS FLIGHT RECORDERS

ISSUED 30 MAY 2002 BY FAI
ON BEHALF OF THE INTERNATIONAL GLIDING COMMISSION

Glossary.

DOP, add: DOP can be estimated in various ways, including, HDOP (Horizontal position), GDOP (Geometric), PDOP (Position, overall), TDOP (Time) and VDOP (Vertical position). EPE also varies with constellation position. Some definitions from RTCA sources are given below. (AL5)

GDOP - Geometric Dilution of Position - The ratio of position error of a multilateration system (see definition of Multilateration). More precisely, it is the ratio of the standard deviation of the position error to the standard deviation of the measurement errors, assuming all measurement errors are statistically independent and have a zero mean and the same standard distribution. GDOP is the measure of the quality (sometimes, "goodness") of the geometry of the multilateration sources as seen by the observer; a low GDOP is desirable, a high GDOP undesirable. (See also PDOP, HDOP and VDOP.)

HDOP - Horizontal Dilution of Position - The ratio of user-referenced horizontal position error to measurement error of a multilateration system. (See GDOP for a more detailed description.)

PDOP - Position Dilution of Position - The ratio of user-referenced three-dimensional position error to measurement error of a multilateration system. PDOP is the root-sum-square of HDOP and VDOP.

VDOP - Vertical Dilution of Position - The ratio of user-referenced vertical position error to measurement error of a multilateration system (see GDOP for a more detailed description).

add: EUROCAE, European Organization for Civil Aviation Equipment. The European counterpart of the US RTCA, see under RTCA. It has fewer members than RTCA but carries out joint projects. National aviation authorities often refer to RTCA/EUROCAE documents. (AL5)

FR Serial Number, first sentence, delete words after the comma. At the end, add: If complete and unambiguous identification is required, the s/n shall be prefixed by the manufacturer's name and the FR model number. (AL5)

add: Multilateration - literally, having many sides, the shape of a geometric figure formed by nearly-intersecting lines-of-position. Therefore, in navigation systems, the obtaining of a fix from multiple lines of position. (AL5)

add: Pseudorange - a measure of the apparent propagation time from the satellite to the receiver antenna, expressed as a distance. The distance from the user to a ranging source (for instance a

satellite) plus an unknown user clock offset distance. With four ranging source signals it is possible to compute position and offset distance. If the user clock offset is known, three ranging source signals would suffice to compute a position. (AL5)

add: RTCA Inc - a US not-for-profit corporation concerned with aviation and aviation electronic systems. It functions as a US Federal Advisory Committee and develops recommendations on aviation issues based on consensus. Nearly 300 organizations are members, about one-quarter being non-US, one of which is FAI. The initials RTCA originally stood for Radio Technical Commission for Aeronautics, a previous government body which was discontinued. The initials RTCA no longer have a longer official form, but these words appear on the bottom of its letter pages in italics: Requirements, Technology and Concepts for Aviation. (AL5)

2.6.3 **Memory**. Add: When the memory is full, flight data shall continue to be recorded, overwriting the earliest data previously recorded rather than stopping recording. (AL5)

2.6.5 **Pressure altitude sensor**. Add after the first sentence: "The pressure altitude measurement system must be compensated for temperature changes. The system will be tested by GFAC for accuracy and freedom from anomalies in recording pressure altitude, and the sensor must be of a type that is acceptable to GFAC." (AL5)

2.7.1 **GNSS FR Equipment Requirements**. Add: "The type of GPS receiver board used will be tested for accuracy and freedom from anomalies in recording lat/long and altitude, and must be of a type that is acceptable to GFAC ." (AL5)

2.7.2.1 **Type of seal**. Change to: The enclosure must have a tamper-proof physical seal across a joint or screw, so that the seal will be broken if the case is opened. For the internal security mechanism, see para 2.8.4. (AL5)

2.7.2.2.2 **Antenna**. Second sentence to start: "The antenna, its wiring and connection to the FR are critical parts of the system without which GPS fixes will not be obtained. In case of a defect, the components external to the FR case must be easy to replace with serviceable ones. Therefore, for FRs with external antenna connections (it is recognised that some designs have antennas that are integral with the FR case and an external connector does not apply), any antenna connector on the FR case must be to a type of antenna cable connector that is commonly-available worldwide, be separate from other connectors and be designed". Also, the third sentence to read: "Such as the BNC bayonet antenna connector; a bayonet connector being generally preferred over a screw connector because the latter can become gradually unscrewed with usage, unless secured by locking agent."

2.7.2.2.6/7 **Connectors at the PC end**. Make provision for USB ports on PCs. 2.7.2.2.6 and 7 to read:

2.7.2.2.6 **Connector cable for data transfer from FR to PC**. The types of IGC-approved connectors on the FR case are specified in 2.7.2.2.7. The PC is expected to have a 9-Pin RS232 male connector for data input unless it only has USB connectors, in which case see 2.7.2.2.6.2. If the FR case has the IGC-approved type of RS232 connector, no special extra cable is required to be supplied by the manufacturer. If the FR case has an IGC-approved connector other than the

RS232, the FR manufacturer shall provide a cable which connects to the FR at one end and has a female RS232 connector on the other end. This RS232 will either be connected directly to the PC or, for PCs with only USB connectors, via an interface connector provided by the pilot, OO, or the owner of the PC. (AL5)

2.7.2.2.6.1 IGC standard cable for data transfer. It is expected that all persons involved in transferring data from FR to PC will have access to a cable for data transfer with RS232 connectors, male for the FR end and female for the PC end. This shall have straight-through wiring, for instance, female pin 2 is connected to male pin 2, and so forth. 'Cross-Over', 'Null Modem', Gender Changer or Manufacturer-Specific cables must not be used because their wiring does not conform with the IGC standard and their use could damage the FR or the PC, or lead to loss of data. (AL5)

2.7.2.2.6.2 USB connectors. For PCs with USB connectors and without a RS232 port, a conversion cable from USB to RS232 must be supplied, normally by the owner of the PC, so that it can be used with the IGC connectors and cables specified above. With the trend towards USB ports, this will be kept under review, but provision needs to be made for the use of older PCs as well as new ones and it is anticipated that RS232 PC connections will be part of this Specification for some time. (AL5)

2.7.2.2.7 Connectors on the FR case for data transfer. IGC-approved types of connectors on the FR case are listed below, one of which shall be fitted to all stand-alone FRs. Recorders that could be panel-mounted must also use one of the IGC-approved connectors. Because in some installations the FR case in a panel-mounted system may be difficult to access, an extension cable must be supplied so that one of the IGC-approved connectors is always readily available for connecting a PC for data transfer. When the recorder is fitted to the instrument panel, the manufacturer's instructions must make the point that the IGC standard connector on the extension cable must be available in a convenient place in the glider for transferring data to a PC, if necessary by use of a further extension cable to the PC itself. (AL5)

2.7.5 Unique Serial and Version Numbers. Last sentence to be replaced by: "The case of the recorder must be permanently marked with the name of the recorder manufacturer, type/model and S/N. Where a display is available, the same data must be shown on that display for at least 10 seconds before it changes to another presentation, so that it can easily be written down by an OO before flight. Other details on the display are optional because they are included in the IGC file header record. Where a recorder is designed to be mounted in an instrument panel, the data above must be easy to see and record from the cockpit without having to look at or make adjustments to the back of the panel. The required data must be available to an OO without having to access the back of the panel. For instance be on the display after switching on or by permanent marks on the front of the case including inside the instrument display itself. (AL5)

Replace paras 2.8.1 and 2.8.2 with: 2.8.1 Levels of IGC-approval. (Based on SC3B para 1.1.3.3) The IGC-approval document for individual types of Flight Recorders will specify procedures to be used and any limitations on types of flights for which the approval is valid. Reduced levels of approval apply to types of Flight Recorders that do not meet the requirements for full approval at the time that the approval is given. They also apply in cases where, for whatever reason, the security of a type of recorder may have been compromised, or is judged by GFAC to be not

suitable at the higher security level, or been determined by GFAC to be below the requirements of the current Specification for certain types of flights. The following levels of IGC-approval apply, in addition to no approval or withdrawal of an existing approval: (AL5)

2.8.1.1 Full IGC-approval. This applies to Flight Recorders that may be used for evidence for all flights up to and including FAI/IGC world records.

2.8.1.2. IGC-approval for badge flights up to Diamonds. This applies to Flight Recorders that may be used only for evidence for Silver, Gold and Diamond badge flights. They may have lower standards of security compared to Flight Recorders with higher levels of approval. This level also includes systems that use a separate GNSS unit connected to the Flight Recorder unit by cable.

2.8.2. **Recorder units connected by cable to a separate GNSS receiver**. For FRs approved for use only for Silver, Gold and Diamond Badge flights, IGC will permit recorders with separate GNSS receiver units to be used subject to the other provisions of this Specification and Sporting Code Section 3. In particular they must have the ability to generate the IGC format flight data file and have a standard of physical and electronic security appropriate to badge flights up to Diamond, as determined by GFAC. (AL5)

2.8.2.1 Higher level approvals. Before an approval to a higher level than Diamonds can be considered, the security of electronic flight data from the GNSS unit to the FR must be to a similar standard as that required for such data which is transferred from the FR unit itself to storage media. A separate receiver must be securely physically sealed, and the data stream from the GNSS receiver to an FR unit must contain both an electronic identification of the individual receiver unit (so that in the event of a query on the data, it can be identified and inspected), and also a digital signature (DS) system no less secure than that required for data from the FR unit itself. (AL5)

2.8.3.1.1 **RSA, DSA and data transfer times**. 6th sentence to read: A 100kB IGC data file must transfer with full security encoding from the FR to a Pentium 100 PC in 1 minute or less. (AL5)

2.8.4. **Unauthorised Changes**. Change title to "Changes "from "Interference". The latter could be confused with electronic interference. Second sentence, change "can" to "shall".

Also, add: If the internal security mechanism (for instance a microswitch) has operated, the recorder shall continue to produce flight data in the IGC file format, although such data must fail the various security tests and cannot be used for flights that require validation to IGC/FAI criteria.

Authorised agents of FR Manufacturers or anyone else shall not be given (or acquire over time) lists of codes or keys for re-initialisation purposes, that could be used again without reference to the FR Manufacturer. The system must be protected in the case where an agent might transfer to another manufacturer or even be involved in hacking at a future date. The normal procedure shall be that a security code or key is transmitted by the FR manufacturer to an authorised agent for the re-sealing of an individual recorder on a specific date, the code or key being itself coded or using a "one-use pad" system. This is so that an agent or anyone else cannot, over time, collect a database that would allow recorders to be initialised without reference to the FR manufacturer and that could be used later in some illicit way. The use of publicly known algorithms is preferred, so that the level of security can be assessed by GFAC. Also, to prevent loading of invalid programs, if

application code with an invalid signature is loaded, the battery backed RAM must be cleared. (AL5)

2.9. **Short Program Files** after the heading, insert: This refers to the DATA, COM and VALI functions and short program files. These file programs will be made available by IGC as freeware on FAI internet reference: ftp://www.fai.org/gliding/software/gps/pc (also through a link from http://www.fai.org/gliding/gnss). Manufacturers shall check from time to time that their program files on these web references are up to date. FR manufacturers must also distribute them as part of their normal software package to customers and their literature and manuals shall remind customers to check the web site above from time to time in case updated versions have been made available. (AL5)

2.9.3 **Short Program Files for Transfer and Conversion.** Add a new sentence after the second sentence: "... operating system of the PC. For PCs using one of the MS Windows OS, the short program files must be capable of executing correctly either through a self-booting floppy disk, or from the main Hard Disk with Windows running, if necessary through the DOS window (where one exists in the OS concerned). Where MS Windows-based equivalents of these programs are produced (see Appendix 3 to this Specification), both DOS and Windows programs will be placed on the gliding/gnss web pages for free general use for the recorders concerned. (AL5) "

Also in 2.9.3, remove the double indent in the line that begins: "-n [file]" and replace with a single indent so that it is in line with the other switches above.

2.9.3.1 **Data transfer from FR to PC - the DATA file.** Replace the existing 2.9.3.1 with the following:

2.9.3.1 **Data transfer from FR to PC - the DATA file.** The program file DATA-XXX.EXE shall transfer the flight data from the FR memory to a PC and shall also convert transferred data to the IGC file format at the same time. The DATA file shall be as small as possible, must work alone (not depending on any other files or programs), and is not expected to exceed 200 kb. Similar programs which will work on Mac computers may also be produced, but where evidence is submitted to NACs and FAI on portable disks, the disk must be formatted for use on PCs, although email is now normally used and the IGC file can be sent as an attachment. Appendix 1 para 1.1.1 gives the conditions for which the DATA program and the FR must produce a separate IGC flight data file. The DATA program file shall be the shortest software program which can transfer data from the FR to a PC and can help to ensure that data is in the hands of an OO at the earliest possible time after flight. Many DATA programs for different FRs can be held on a PC (or on a floppy diskette) at the gliding club, other organisation at the airfield, or, for championships, in competition control. For those FRs that have a more comprehensive manufacturer's software program in addition to the short program files, the DATA file provides a free and rapid option for data transfer in the absence of other programs. (AL5)

2.9.3.1.1 **Last flight.** On executing the DATA-XXX program file, the data from the last flight must automatically be transferred to the PC to the same directory as the DATA file, without needing special switching or keyboard actions. Transfer of the data on the last flight shall be as quick as possible (at competitions up to 100 FRs may be involved and long time transfers cannot be tolerated for individual IGC files). (AL5)

2.9.3.1.2 Options through menu. After the automatic transfer of the last flight, options must be offered through a menu for transferring data from other flights in the FR memory. The menu shall list all flights in the FR memory in reverse chronological order, giving the date/times of start and finish of recording and the duration of the recording. After the automatic transfer of the last flight, it must be possible to scroll to and highlight other flights for downloading either individually or collectively. If a selected flight is already in the PC memory, it shall be possible to bypass the further downloading of that particular flight. All erase and overwrite actions must have a second check keystroke to avoid inadvertent erasure or overwriting of data already in memory in the FR or the computer. (AL5)

2.9.3.1.3 Execution of program. The DATA-XXX program file must transfer data when executed and connected to an FR which is switched on. No special setting up of the FR itself shall be required before data transfers, although a timeout function of at least 60 seconds is acceptable after the FR is switched on during which contact with a PC may be made. A 100kB IGC data file must transfer with full security encoding from the FR to a Pentium 100 PC in 1 minute or less. (AL5)

2.9.3.1.4 Ports. The DATA-XXX program must be capable of downloading via serial COM ports 1 and 2, and manufacturers shall include compatibility with other COM ports, LTP ports, USB ports, and shall allow (wherever possible) for any future commonly-available ports. For COM ports, only the port number is required; for other ports the three-letter designation of the port type plus the port number is required. For Example: DATA-XXX -pLPT1, or /pUSB2. (note the space between the file name and the switch). The DATA, COM and VALI files will be made available as freeware on the FAI internet reference: <http://www.fai.org/gliding/gnss/freeware.asp> (also through a link from <http://www.fai.org/gliding/gnss>), and manufacturers shall check from time to time that their program files are up to date. FR manufacturers must also distribute them as part of their normal software package to customers. (AL5)

2.9.3.1.5 Integrity of data. Where a problem is detected in the integrity of the recorder or the data (such as if the recorder has been opened in an unauthorised manner) and the recorder has a screen display, the words describing the problem must stay on screen until positive action is taken to proceed to the next stage in the data transfer process. This is so that the person viewing the screen can take notice of the message, even if distracted when the message first comes up. (AL5)

2.9.3.4 **Digital Signature (DS) Check Program - the VALI file.** This shall indicate a positive result when executed. Insert two sentences towards the end of this para before "NACs may copy ...": "The result shall be presented clearly on screen, either pass or fail. If "pass", the words "Validation check passed, data indicated as correct" shall appear. If "fail", after "Validation check failed", a likely reason shall be given if possible, for instance "File data may have been altered", or "security microswitch may have operated, recorder case may have been opened". The result must remain on screen until a positive action is take to change it." Also, add at the end of 2.9.3.4: "Validation software should also work correctly if the CRLF at the end of a line is replaced by CR or LF. This will make it possible that a file transferred in ASCII

mode from a MS operating system to a UNIX system and in binary mode from a UNIX system to a MS operating system, will continue to pass the VALI-XXX check. (AL5)"

2.9.3.5 **Multiple Programs - File Names.** Words in brackets, replace "becomes" by "is followed by".

2.11.1 **Engine Noise Level (ENL) systems.** Add after the first sentence: "Systems must be capable of utilising all of the numbers between 000 and 999."

Appendix 1, 1.1.1.1 **FRs that do not start recording fixes immediately after switch-on.** Change title to "Thresholds for starting and stopping recording, also Power-on Protocol for creating new IGC files". Also, third sentence to be placed at the end and to read: "After landing, this can be achieved by a time delay before stopping the recording of fixes. Since this condition may also apply to a period in wave or ridge lift with little or no vertical or horizontal movement, a new IGC file must not be started while power is still applied, even when no movement is being detected and no fixes are being recorded." Note that this "power-on protocol" is also needed for barograph calibrations so that a new file is not produced while the calibrator is making small adjustments to the pressure at each level of the calibration. A new IGC file is produced under the "Power-off Protocol" that is described under 1.1.1.2 below. (AL5)".

Appendix 1, 1.1.1.2. **Loss of external power for short periods.** In the title, add "- Power-off Protocol for creating new IGC files". Also, in line 2, replace "should" by "must".

Appendix 1, 2.5.6. **Manufacturer codes.** Revised paragraph:

2.5.6 **Manufacturer codes.** Single- and three-character codes are tabulated below. Manufacturers applying and paying the fee for IGC-approval who are not listed should apply to the Chairman of GFAC for allocation of codes. Manufacturers using the IGC file format but not applying for IGC-approval should use the X and XXX codes, see note 2 at the end:

IGC DATA FILE FORMAT - CODES FOR MANUFACTURERS OF IGC-APPROVED FLIGHT RECORDERS (AL5)					
Single Char. Code	Three Char. Code	Manufacturer Name	Single Char. Code	Three Char. Code	Manufacturer Name
A	GCS	Garrecht	S	SDI	Streamline Data Instruments
B			T		
C	CAM	Cambridge	U		
D			V		
E	EWA	EW Avionics	W	WES	Westerboer
F	FIL	Filser	X	XXX	Other manufacturers, see note below

G			Y		
H	SCH	Scheffel	Z	ZAN	Zander
I			0		
J			1		
K			2		
L	LXN	LX Navigation	3		
M			4		
N			5		
O			6		
P	PES	Peschges	7		
Q			8		
R	PRT	Print Technik	9		

Notes:

1. Where possible, the initial letter of a manufacturer's name will be allocated for the single-character code. Where this letter is already allocated, the next letter in the manufacturer's name will be used, and so forth.
2. X and XXX are general designations for IGC format files from manufacturers who do not produce an IGC-approved recorder. Such recorders will not have been tested and evaluated by GFAC and may not fulfil certain aspects of the IGC Specification such as security protections, recording of pressure altitude, ENL or other variables. There is no guarantee that the file will conform exactly to the IGC format, although specimen files will be checked for compliance with the IGC format if emailed to the GFAC chairman for evaluation. Even after this procedure, no compliance guarantee can be made because the type of recorder will not have completed a full GFAC evaluation. It should be noted that the details of the manufacturer and the recorder model concerned will be identifiable (if the file conforms to the IGC standard) because they will be included in the H (Header) record, see below under H Record in the line:
HFFTYFRTYPE:MANUFACTURERSNAME,FRMODELNUMBER CRLF
3. The three-letter codes PFC, PLT and OOI will not be used since they could cause confusion in the L record.

Appendix 1, new 2.5.7 for a summary of mandatory file records:

Certain records are mandatory for any IGC file. They are summarised as follows:

Record type in file order	Appendix 1 reference	Remarks
A	3.1	Manufacturer and unique ID for FR
H	3.3.1	Header record
I	3.4	FXA extension is mandatory
B	4.1	Fix record (lat/long/alt etc.)
F	4.2	Satellites used in B records
G	3.2	Security record

Appendix 1, 3.3.1. **Header records**.

Change the note after "Data Source" to: Use F if the data is transferred from the FR even though it was originally entered by a pilot or OO. This is to ensure that data that is entered before flight is protected by the security system in the recorder, and if it is changed after flight it will fail the VALI check.

Also, add:

HF FRS SECURITY SUSPECT USE VALI PROGRAM: TEXTSTRING CRLF

Notes: Only to be used where security is suspect and is detected as part of the DATA program. The words before the colon are mandatory. The reason for referring to the VALI program is that this is the only complete way of carrying out a full security check. This small element in some DATA programs must not be seen as a replacement for the full VALI check, which also detects any changes after data transfer. The text string shall be a description of the likely fault, such as: "Security Microswitch operated, recorder needs to be re-set"

Appendix 1, 3.4. **I Record**. Sentence about FXA, break into two as follows: "The Fix Accuracy (FXA) must be included, in the form of the Estimated Position Error figure (see Glossary under EPE). This shall be followed by SIU and ENL if these are recorded in the FR concerned."

Appendix 1, 4.1. **B Record**. First sentence, change words in brackets to "(FXA, in the form of the Estimated Position Error figure, see Glossary under EPE)".

Appendix 1, 4.5. **L Record - Log Book/comments**. Add at the end: "In the case of Free Flights where waypoints are claimed post-flight by the pilot, the PFC (Post-Flight Claim) code shall be used followed by the Waypoints in the same format as the C Record (Pre-flight declaration). It is expected that this data will be inserted by a software program (for instance, after transfer of the flight data to a PC immediately after flight, by a program outside the FR) rather than added manually, so that it is in the correct format and can be recognised by analysis programs that are designed to read Post-Flight Claim data and present the flight on screen together with the PFC waypoints. Note that L records when originated by a pilot or OO do not affect the VALI check for the rest of the file data.

Add to examples: "L PFC TEXT AS C RECORD CR LF"

Add similar line to the table.

Appendix 1, 5. **Definitions** FR Serial Number, delete last two sentences.

Appendix 1, 7. **Three-letter Codes (TLC)**. Add:

"PFC - L - Post-Flight Claim. For Free Flights where waypoints are claimed post-flight.

FRS - H - Flight Recorder Security. Where the DATA-XXX program has detected a security fault such as the recorder security microswitch having operated."

Appendix 2 **Tests for security**. Para 5, Anti-tampering Protection, insert new 5.3:

"5.3. Evaluation and tests will be made to determine that the electronic and physical security of the FR has been provided with protection to minimize the possibility that a determined attempt to bypass the security features will succeed. For

instance, any security microswitch must be undefeatable (as far as is physically possible), being fitted in such a position (for instance with shields or guards where necessary) to protect against the insertion of a specially shaped tool into the case of the recorder which might hold down the microswitch while the case is being opened, thereby allowing unauthorised design changes to be made without the security mechanism being activated. GFAC will open the cases of types of recorders that are under test, evaluate these features and require changes where these are deemed to be necessary to preserve security in subsequent worldwide operation of the type. (AL5)"

Appendix 3 - Windows-based programs, revised Appendix 3.

Appendix 3 to
Technical Specification for
IGC-approved GNSS Flight Recorders (AL5)

Windows-based short programs
for data transfer and conversion to IGC format,
and validation of data

1. **INTRODUCTION.**

1.1 **General.** This appendix describes an IGC standard for Manufacturer-supplied Windows Dynamic Link Libraries (DLLs) in a 32-bit Windows environment, for the DATA, CONV and VALI functions that are described in para 2.9 of the main part of this Specification.

1.2 **Availability on the IGC GNSS web site.** The DLLs shall be freeware and be made available through links from the IGC/GNSS web site <http://www.fai.org/gliding/gnss/freeware.asp>. IGC will supply a sample control program, in both source and executable form. This will load a DLL and call each of the Application Programming Interface (API) functions specified below. This control program will also be available through links from the IGC web site.

1.3 **Functions supported.** All such DLLs shall support the functions described in the API below.

1.4 **Control program.** A Control Program must check responses from the DLLs and also perform checks for the existence of a file before calling on a DLL to open it for reading (e.g. ConvertLog or ValidateLog) and to query overwriting an existing file (e.g. DownloadLog and ConvertLog). The Control Program shall also select the COM port to be used.

1.5 **DLL Naming.** The name shall be of the form IGC-XXXy.DLL where XXX is the Manufacturer's three Letter Code as defined in Appendix 1 para 2.5.7. A manufacturer's DLL shall be able to handle all GNS FRs in the product range concerned. In case this cannot be achieved, the symbol "y" above is an optional alphanumeric for other DLLs from a given manufacturer.

1.6 The provision of these DLLs is not a mandatory requirement at this time. However, Flight Recorder Manufacturers are strongly advised to provide such DLLs, because this is expected to become a requirement in the future. This is to allow for problems encountered running the 'Short DOS programs' on some PCs, and may replace the requirement for DOS programs in a future edition of this document.

1.7 If problems are encountered by any FR manufacturer in implementing a DLL due to the provisions of section 2 of this Appendix, this shall be reported to GFAC chairman as soon as possible. Changes from section 2 will not normally be considered unless the request is due to hardware characteristics of a Flight Recorder approved before the publication of amendment 5. This provision is likely to be withdrawn in the next amendment so manufacturers are urged to ensure that they can comply in due course.

2. **APPLICATION PROGRAMMING INTERFACE (API) FOR MANUFACTURER-SUPPLIED DLLs**

2.1 **General.** A standard API is described below in para 3. It includes the functions mentioned in para 1.1 for the operating systems Windows 95/98/ME, NT/2000, and subsequent releases. The API shall be implemented by a DLL supplied by each flight recorder manufacturer, which exports a defined set of functions for use by control programs. These control programs may include third-party flight evaluation applications, competition scoring software, and generic programs for the use of pilots, official observers and contest directors.

2.2 Control programs. The DLLs shall be designed to be stored in a common directory on the computer of the end-user. The control program shall use the Win32 *LoadLibrary* or *LoadLibraryEx* functions to load a DLL using run-time linking. The control program shall then query it for the entry point of each API function by name, using the Win32 *GetProcAddress* function.

2.3 Functions and descriptions. In the API descriptions below, DWORD, BOOL, TCHAR, LPTSTR, and LPCTSTR are standard Win32 API types defined as long, bool, char, char *, and const char *, respectively, for the required ANSI (as opposed to UNICODE) DLL build. HWND is a 32 bit window handle. FALSE is integer 0, TRUE is any non-zero integer value.

3 API SUB-ROUTINE DESCRIPTIONS. The standard API follows. Titles of main sub-routines are in bold, underlined, and end in the letters DLL, FR or LOG. They are followed by relevant data such as parameters, return values, and remarks.

3.1 IdentifyDLL

DWORD IdentifyDLL(LPTSTR value, DWORD size)

The *IdentifyDLL* function obtains an identifying string, which the control program shall enter in a listbox used to select the appropriate DLL.

Parameters

value

[out] pointer to buffer to receive string.

size

[in] size of the buffer pointed to by *value*.

Return Values

Function returns number of bytes in returned string, if actual length of string exceeds *size*, the string shall be truncated to *size* -1 bytes.

Remarks

The string consists of six fields, separated by the "pipe" character ("|", 0x7C) , the manufacturer three letter code and optional alphanumeric, the manufacturer name, supported FR name(s), DLL software revision number, and two comma separated lists of zero or more file extensions. The first list of extensions identifies manufacturer proprietary log files, if any, which can be converted to IGC format using the *ConvertLog* function. The second list identifies log files (possibly including IGC format) that can be authenticated by the *ValidateLog* function. A terminating NUL character is always appended to the string (but not included in the returned count). Maximum permitted length of the string (excluding the terminating NUL character) is 127 characters. Example:

XXX|Acme Instruments|XL 100, 200|2.0|XL1,XL2|XL1,XL2,IGC

3.2 LoadIconDLL

HICON LoadIconDLL()

The control program calls *LoadIconDLL* to load a unique 32x32x4 (16 color) icon that may be used to identify the DLL.

Return Values

Returns the handle for the loaded icon. If there is an error, returns a null handle.

InitializeDLL

void InitializeDLL(HWND windowHandle, BOOL quietMode)

3.3 **InitializeDLL** This is an initialization function that must be called before any of the other functions, with the exception of *IdentifyDLL* and *LoadIconDLL*.

Parameters

windowHandle

[in] Handle for the control programs main window, or NULL if there is none.

quietMode

[in] if TRUE, application is operating in quiet (non-interactive) mode.

Remarks

The window handle shall be stored in the DLL, and is normally used as the parent handle for any dialog boxes displayed by DLL functions, unless overridden by a call to *SetWindowDLL*. These dialogs shall be centered within the parent window. If quiet mode is requested, the DLL functions shall only display dialogs resulting from non-recoverable error conditions, progress and informational dialogs shall not be displayed.

3.4 **SetWindowDLL**

VOID SetWindowDLL(HWND windowHandle)

The control program shall call this function to set a new top-level window handle. This handle shall be used as the parent window for any dialog boxes displayed by DLL functions, unless overridden by another call to *SetWindowDLL*.

Parameters

windowHandle

[in] Handle for the control programs new top level window.

3.5 **KeepAwakeIntervalDLL**

DWORD KeepAwakeIntervalDLL()

The *KeepAwakeIntervalDLL* function is used to obtain the nominal time interval between calls to *KeepAwakeFR*.

Return Values

Returns the interval in milliseconds. If 0 is returned, *KeepAwakeFR* calls are not required (and shall be ignored).

3.6 **UseSerialOptionsDLL**

BOOL UseSerialOptionsDLL()

The *UseSerialOptionsDLL* function is called by the control program to determine if there are any user settable connection options which may be set using the dialog provided by *SerialOptionsDLL*. If there are none, the control program shall not call *SerialOptionsDLL*.

Return Values

Returns TRUE if there are user settable serial options (and *SerialOptionsDLL* shall be called), FALSE if not

3.7 **SerialOptionsDLL**

DWORD SerialOptionsDLL(LPTSTR options, DWORD size)

The *SerialOptionsDLL* function displays a modal dialog box requesting any user settable connection options (line speed, flow control, etc.) needed to configure a serial port for use with the manufacturers FRs.

Parameters

options

[out] pointer to buffer which shall receive the connection options.

size

[in] size of the buffer pointed to by *options* in bytes.

Return Values

If successful, function returns number of bytes in the returned option string. if actual length of string exceeds size, the string shall be truncated to size -1 bytes. If cancelled, returns 0 and options string is left unmodified. If error, a modal dialog is displayed, and -1 is returned.

Remarks

This dialog shall not include selection of the serial communication device. Maximum permitted length of the returned string (excluding the terminating NUL character) is 63 characters. The string is intended for use in a subsequent call to *SerialConnectFR*, the actual format of the string is determined by the manufacturer. The control program may choose to store this string in the registry or a file for use in future sessions.

3.8 **SerialConnectFR**

BOOL SerialConnectFR(LPCTSTR device, LPCTSTR options)

The *SerialConnectFR* function is used to establish communication with a FR connected through a serial port. Must be called prior to using *KeepAwakeFR*, *IdentifyFR*, *IdentifyLogFR*, *DownloadLogFR*, and/or *DisconnectFR*.

Parameters

device

[in] name of the serial communication device device ("COM1", etc.).

options

[in] string returned by a previous call to *SerialOptionsDLL*, or NULL to use the default device options. The format of this string is determined by the manufacturer.

Return Values

Returns TRUE if connection established, FALSE otherwise.

Remarks

If a connection cannot be established, the function shall display a modal dialog box detailing the problem.

3.9 **KeepAwakeFR**

BOOL KeepAwakeFR()

The *KeepAwakeFR* function is used to prevent the FR from disconnecting during idle periods between calls to *ConnectFR* and *DisconnectFR*. If *KeepAwakeIntervalDLL* returns a non-zero value, the control program must call *KeepAwakeFR* each time that interval elapses.

Return Values

Returns TRUE if the FR still connected or FALSE if connection has been broken.

3.10 **IdentifyFR**

DWORD IdentifyFR(LPCTSTR value, DWORD size)

The *IdentifyFR* function is used to obtain the manufacturer id/serial number, the FR model name/number, and the FR sealed status for the connected FR.

Parameters

value

[out] pointer to a buffer which shall receive the string result.

size

[in] size of the buffer pointed to by *value* in bytes.

Returned Values

Function returns number of bytes in returned string, if actual length of string exceeds *size*, the string shall be truncated to *size* -1 bytes.

Remarks

The string consists of three fields, separated by the "pipe" character ("|", 0x7C), the manufacturer id/serial number (formatted MMMNNN, where MMM is the manufacturer id, and NNN is the serial number), the FR model name/number, and the FR sealed status ("SEALED" if sealed, "UNSEALED" if not). Maximum permitted length of the string (excluding the terminating NUL character) is 63 characters. Example:

AXL01F|XL 100|SEALED

3.11 **IdentifyLogFR**

DWORD IdentifyLogFR(DWORD index, LPSTR value, DWORD size)

The *IdentifyLogFR* function is used to obtain information on a log stored in the currently connected FR.

Parameters

index

[in] index of the desired log, starting with 0.

value

[out] pointer to the buffer which shall received the returned string.

size

[in] size of the buffer pointed to by *value*.

Return Values

IdentifyLogFR returns number of bytes in the returned string, if actual length of string exceeds *size*, the string shall be truncated to *size* -1 bytes. If the value specified for *index* exceeds the number of logs present in the FR (minus 1, as indexing starts with 0), *IdentifyLogFR* shall return 0.

Remarks

The returned string consists of seven fields, separated by the "pipe" character ("|", 0x7C), the default log file name (including extension), log start UTC date (formatted YYYY-MM-DD, example "2000-05-12", zero padding required), log start UTC time (formatted HH:MM:SS, example "17:09:22", zero padding required), log end UTC time (formatted HH:MM:SS), pilot name, competition id, and competition class. Maximum permitted length of the returned string (excluding the terminating NUL character) is 127 characters. Example:

0B8X01F1.XL1|2000-11-08|20:05:21|01:21:09|J. Doe|XYZ|15M

Logs are indexed in descending start date/time order, the log at index 0 is the most recent log. When retrieving information on all of the logs stored within the FR, a control program shall start by calling *IdentifyLogFR* with *index* 0, incrementing *index* by 1 until *IdentifyLogFR* returns 0.

3.12 **DownloadLogFR**

BOOL DownloadLogFR(DWORD index, LPCTSTR fileName)

The *DownloadLogFR* function is used to download a log file from the currently connected FR.

Parameters

index

[in] the index of the desired log, starting with 0.

fileName

[in] a null terminated string containing the name of the file (which may include a path) to which the log shall be downloaded. If NULL, the default file name shall be used in the current working directory.

Return Values

DownloadLogFR returns TRUE if successful, FALSE if there was an error.

Remarks

If a file with the specified name and path already existed, it shall be overwritten. If there is an error, *DownloadLogFR* shall display a modal dialog box giving the details. If the quietMode flag was set to TRUE in the call to *InitializeDLL*, the download shall occur silently, without any dialog boxes displayed, unless there is an error. If quietMode was set to FALSE, *DownloadLogFR* shall display a modal dialog box with a progress indicator and a cancel download button.

3.13 **DisconnectFR**

VOID DisconnectFR()

DisconnectFR is called after the control program has completed interaction with the FR, to close the communication device.

3.14 **UseConvertLog**

BOOL UseConvertLog()

The *UseConvertLog* function is called by the control program to determine if the DLL *ConvertLog* function is needed to convert from a proprietary log file format to IGC. If it does not, the control program shall not offer the user the option of converting log files.

Return Values

Returns TRUE if conversion from proprietary format to IGC is required, FALSE if not

3.15 **ConvertLog**

BOOL ConvertLog(LPCTSTR fileName, LPCTSTR igcFileName)

ConvertLog converts the log file specified by *fileName* to an IGC format file specified by *igcFileName*.

Parameters

fileName

[in] a null terminated string containing the name of an existing log file (which may include a path) in the manufacturer proprietary format.

igcFileName

[in] a null terminated string containing the name of the IGC file (which may include a path) to be created.

Return Values

Returns TRUE if successful, FALSE if there is an error.

Remarks

If a file with the specified *igcFileName* already exists, it shall be overwritten. If there is an error, the function shall display a modal dialog with the details.

3.16 **ValidateLog**

BOOL ValidateLog(LPCTSTR fileName)

ValidateLog is called to authenticate the digital signature on a specified log file.

Parameters

fileName

[in] a null terminated string containing the name of an existing log file (which may include a path) to be validated.

Return Values

Returns TRUE if file can be validated, FALSE otherwise.

Remarks

If the log was not produced by a supported flight recorder, is in an unsupported format, or the digital signature is invalid, the function shall display a modal dialog detailing the problem, then return FALSE.

If quietMode was set to FALSE in the call to *InitializeDLL* and the validation takes more than a few seconds, *ValidateLog* shall display a modal dialog box with a progress indicator and a cancel validation button.

4. PROGRAMMING FRAMEWORK FOR CONTROL PROGRAMS

In order to utilise the freeware DLLs described earlier in this appendix, a Control program is required. Authors of Analysis Software, Scoring Software etc. may wish to incorporate the facilities of such a program into their products. In order to assist Flight Recorder manufacturers and other Software writers, the following files are provided on the IGC website at <http://www.fai.org/gliding/gnss/freeware.asp>.

Working Sample Shell Program.

Source of Shell Program.

Dummy DLL file (to assist in Shell Program testing).

Source of Dummy DLL. (can be used as a Template for Manufacturer's DLL code)

These files are provided free of charge and with no warranties of any kind. If portions of the source files are used in any product, then the copyright conditions in the source files are to be observed.

[Note: paragraph 4 is also used on the IGC web site as an introduction to the files for free downloading.]

----- end of amendment -----