

BiSON

Birmingham
Solar-Oscillations
Network

TECHNICAL REPORT NO. 237

BiSON Data-File Formats

John Allison, William J. Chaplin, Yvonne Elsworth, Steven J. Hale, and
Brek A. Miller

The University of Birmingham, Edgbaston, Birmingham B15 2TT

2004 September 15

This technical report series is published by:



**THE UNIVERSITY
OF BIRMINGHAM**

High-Resolution Optical-Spectroscopy Group

School of Physics and Astronomy
The University of Birmingham
Edgbaston, Birmingham B15 2TT, United Kingdom
Telephone: +44-121-414-4551 FAX: +44-121-414-1438

BiSON Data-File Formats

John Allison, William J. Chaplin, Yvonne Elsworth, Steven J. Hale, and
Brek A. Miller

The University of Birmingham, Edgbaston, Birmingham B15 2TT

2004 September 15

Abstract

The RAWDATA, DAT, CMP, SM1, DATA, IZDATA, BBC, SUCIO, and RES file formats are defined.

1 Introduction

The instruments in the Birmingham Solar-Oscillations Network (BiSON) measure the intensity of light scattered from a small, glass cell containing potassium vapor. Photodetectors or photomultiplier tubes are used. These devices output a voltage proportional the intensity of light falling on their active surfaces¹. The output voltage is passed through a voltage-to-frequency (V/F) converter and the resulting pulses routed to a set of counters.

The polarization of the incoming sunlight is controlled by special electronics. The same hardware also sends the appropriate gate signals to the counters. For each point, the counter gates are held open for 3.2 seconds. The gates are then closed for 0.8 seconds to allow the readout electronics to read the accumulated counts from the counters. In this way, one line-of-sight velocity measurement is taken every four seconds.

In order to save CPU time, four-second data collected throughout the day is written directly into the raw-data file. At sunset, the compacting software reads this file, performs some statistical analysis on the data, and writes a daily data file that contains forty-second data.

We will use the following terms when talking about the incoming data:

datum A single value, usually read from a counter.

¹This, and many of the points in the following paragraphs, aren't exactly true for the Izaña system. The differences are noted below in the sections that deal with the Izaña-specific data-file formats.

- point* A collection of *data* all read simultaneously. There are usually four *data* per point, except at stations with lock-in amplifiers, where there are three. The number of data per point increases as things like Delta-B units, separate starboard and port detectors, and magnetic Pockels cells are added to the system.
- set* A collection of *points* that contain data taken in all forms. Basically, this concept was introduced to handle the magnetic-calibration (Delta-B) data. The state of the magnetic-calibration unit changes between *points*. Two consecutive *points* make up a *set*. When the magnetic-calibration unit is not in use, each *set* contains one *point*.
- group* A collection of *sets* that comprise all of the information read in one forty-second block. Each line in the daily data files contains all of the information from one *group*. Either five or ten *sets*, depending on whether a magnetic-calibration unit is or is not being used, are averaged together to make a *group*.

There are several file formats used to store BiSON data:

- RAWDATA Raw-data files. RAWDATA files are written to hard disk while data are being collected. The files are made up of records, each of which contains data from one four-second point. These files are too large to send back to Birmingham and are therefore compacted to DAT and CMP files at sunset each day.
- DAT Daily data files. A DAT file is made up of records, each of which contains data from one forty-second group. Restart records indicate what type of data is being collected. The data are written in ASCII format.
- CMP Compressed daily data files. These files are similar to DAT files except that the data are written in binary format making it easier for programs to read but impossible for humans to read.
- DATA Ten-day data files. These files contain ten days of data collected together into one big file. This format is useful for IBM mainframe computers where the disk-space overhead per file is large, however they are an awful way to store data on any other computer.
- IZDATA Daily data file from Izaña. The files have the type *.dat* making it possible to confuse them with DAT files.
- BBC Another format for daily data files from Izaña. They are also incompatible with DAT files.
- SUCIO Dirty two-second data from Izaña. These files are produced by the bobcat in the zoo system and are converted to RAWDATA format by the panther. Data in RAWDATA files have the time correction applied, data in SUCIO files do not.
- RES Residual files. A RES file is made up of records, each of which contains a residual velocity datum from one forty-second group. A restart record at the beginning of the file indicates what type of data is being collected. The data are written in ASCII format.

2 RAWDATA File Format

Data collected at the remote stations is stored immediately to disk in the raw-data file. This file contains information collected at four-second intervals and, in olden days, was too large to send back to Birmingham by conventional means. Instead, it is compacted into a DAT file at sunset. The compacting routine averages data groups together and writes a DAT file containing forty-second data.

During data acquisition at older QSOOSY stations, the full file specification of the raw-data file is *C:\Results\rawdata.dat*. At new zoo stations, we use a different file specification, it is */home/zoo/Results/ssymmdd.raw* where *ss* is the first two letters of the station name and *yymmdd* is the date on which the data are being collected.

Some QSOOSY sites have Iomega ZIP drives which can store one month of RAWDATA files on a single ZIP disk. When this is done, the RAWDATA files are renamed to *H:\Results\ssymmdd.raw* where *ss* and *yymmdd* are as before.

The RAWDATA file is made up of a series of records. Each line in the file is a record. Under some operating systems (MS-DOS) lines are separated by carriage-return/line-feed pairs. Under other operating systems (Unix) lines are separated by bare line-feeds. Lines may only contain printable ASCII characters.

Each record is a space-delimited list of tokens. There are one or more spaces between each token. There may also be one or more spaces at the beginning and end of each line. Records may not be empty—blank lines are not allowed.

Most tokens are ASCII representations of numeric values and can be either integer or floating point. Floating-point values may be in decimal or scientific notation. Some tokens are string representations of dates.

There are two kinds of records: *restart* and *data*. The first record must be a restart record; generally the rest are data records. However, a restart record is written every time data-acquisition begins or when the type of data being collected changes. So if you stop the computer during the day and restart it, another restart record will be written.

The format of the restart record is:

```
99.999 mm-dd-yyyy n t1 [...]
```

where *mm-dd-yyyy* is the date, *n* is the number of spectrometers, and *t1* is the first data-type bitfield for the data collected by the first spectrometer. Additional data-type bitfields may follow; for example, if more than one spectrometer are in use.

At some stations, data are collected beyond 24^h UT, which means that the date changes during data acquisition. At these stations, a special method is used to determine the date to write in the restart records. The dates in the restart records always corresponds to the times

Table 1: Data-Type Bitfield

<i>Bit</i>	<i>Constant</i>	<i>Data Collected</i>
0	D.CHOPPER	Chopper engaged
1	D.DELTAB	Magnetic-calibration unit in use
2	D.MAG	Magnetic data, no transmission monitor
3	D.LOCKIN	Lock-in amplifiers in use
4	D.NOEOLM	Modulation suppressed
5	D.TWOPOC	Two Pockels cells in use
6	D.STARPORT	Separate starboard and port V/F converters
7	D.MAGCAL	High-speed, magnetic-calibration unit in use
8	D.FOREAFT	Two-magnet data
9	D.PHOTOM	ASP data
10	D.ATTN	Attention, problem at station
15	D.MOREBITS	Another bitfield follows

in the data records and the combination of the two are chosen so that there is no date change or discontinuity in the time value at any time during the day. At some stations, data files will contain times greater than 24^h UT, while at others the data files will contain times less than 0^h UT.

Each bit in the data-type bitfield has a different meaning. Symbolic constants are defined in the software to help you work with data types. Table 1 summarizes the data-type bitfield. This field tells the compacting routine how many data fields to expect in each data record.

We have not yet assigned more than sixteen bits, however provision has been made to accommodate extra bits. The highest-order bit (D.MOREBITS) has been reserved as a marker to indicate that another 16-bit bitfield follows.

Data-analysis programs should silently ignore any set bits that are not understood by the program. It is important to be aware that extra bitfields will be present when D.MOREBITS is set.

The data-type bitfields tell the compacting routine how many data fields to expect in each data record. Generally there are four data fields. If lock-in amplifiers are used, there are only three fields because there is no lock-in amplifier on the transmission signal, hence the transmitted ratio is unavailable. If separate starboard and port V/F converters are used, then there are six data fields instead of four. Bits D.TWOPOC and D.MAGCAL, when set, indicate that the number of data fields is doubled. Both bits D.MAG and D.TWOPOC indicate that magnetic data are being collected. Bit D.MAG means that the data appear in place of the transmission-monitor data, while bit D.TWOPOC means that it appears in addition to the transmission monitor data.

At some older stations, the number of spectrometers is omitted from the restart records.

The remainder of the RAWDATA file is made up of *data* records. There is one record for every four-second *point*, except in Izaña, where points are collected every two seconds.

The data records contain several fields. Each field is either a floating-point value or a long integer. The first field is the time (UT) in hours written as a floating-point value. This value will be in the range -12^{h} to $+36^{\text{h}}$. It will correspond to the date in the most recent restart record.

The fields that follow come in pairs: Each ratio field is followed by a sum field. The scattered-detector pairs are written first, followed by the transmission-monitor pairs. If D.STARPORT is set, a scattered starboard pair precedes a scattered port pair, with one transmission pair following.

The ratios are written as floating-point values while the sums are written as long integers. When lock-in amplifiers are used, the base number of fields is three, because there is no transmitted ratio.

When other bits are set, additional fields appear in the data records. It is possible to find a set of rules that determine the order of the data for arbitrary combinations of data-type bits, however it is rather complicated. This is because new types of data were invented for each new station as they were built. Each scaler system was designed individually — there was no master plan or predetermined standard to follow.

Therefore, the best way to describe what data are present in each column of the data files is to go through each station, one by one. In the tables that follow, special symbols are used to represent the type of data in each column of the data file. The symbols will look something like this: ${}^{\text{F}}\text{R}^{\text{+}}$. The character in the center of the symbol will be R for *ratio* or S for *sum*. The left superscript, if present, will be F for *fore* or A for *aft*. The left subscript will be S for *starboard* or *scattered*, P for *port*, and T for *transmitted*. The right subscript will be + or – to indicate the state of the magnetic Pockels cell. The right superscript will be + or – to indicate the state of the fast magnetic-calibration (Delta-B) unit.

The state of the ordinary magnetic-calibration (slow Delta-B) is not shown because different states are never in the same RAWDATA-file record. The slow Delta-B changes state between points. When D.DELTAB is set, alternate records in the RAWDATA file corresponds to the same slow Delta-B state.

3 DAT File Format

The daily data are transferred from the remote stations to Birmingham either across a modem link or via the Internet. The data are sent as DAT files. These are ordinary text files and they have a name of the form *snymmdd.dat* where *s* is the first letter of the station; *n* is the second letter of the station name, or “b” for the second instrument at a station; and *yymmdd* is the date on which the data were collected. DAT files are written to the *C:\Results* directory on the station computers running QSOOSY and to the */home/zoo/Results/* directory on zoo computers.

The format of DAT files is very similar to that of the RAWDATA files.

Table 2: Data Columns for Klaus in Mount Wilson*

$\overset{\text{F}}{\underset{\text{S}}{\text{R}}^+}$	$\overset{\text{F}}{\underset{\text{S}}{\text{S}}^+}$	$\overset{\text{F}}{\underset{\text{P}}{\text{R}}^+}$	$\overset{\text{F}}{\underset{\text{P}}{\text{S}}^+}$
$\overset{\text{F}}{\underset{\text{S}}{\text{R}}^-}$	$\overset{\text{F}}{\underset{\text{S}}{\text{S}}^-}$	$\overset{\text{F}}{\underset{\text{P}}{\text{R}}^-}$	$\overset{\text{F}}{\underset{\text{P}}{\text{S}}^-}$
$\text{T}\overset{\text{R}}{+}$	$\text{T}\overset{\text{S}}{+}$	**	**
$\overset{\text{A}}{\underset{\text{S}}{\text{R}}^+}$	$\overset{\text{A}}{\underset{\text{S}}{\text{S}}^+}$	$\overset{\text{A}}{\underset{\text{P}}{\text{R}}^+}$	$\overset{\text{A}}{\underset{\text{P}}{\text{S}}^+}$
$\overset{\text{A}}{\underset{\text{S}}{\text{R}}^-}$	$\overset{\text{A}}{\underset{\text{S}}{\text{S}}^-}$	$\overset{\text{A}}{\underset{\text{P}}{\text{R}}^-}$	$\overset{\text{A}}{\underset{\text{P}}{\text{S}}^-}$

*The data-type bitfield contains the following bits: D.STARPORT, D.MAGCAL, and D.FOREAFT resulting in a decimal value of 448.

**The scaler card connected to the transmission monitor contains four counters, like all of the other scaler cards. However, only the first two count meaningful data. The acquisition software discards the data from the other two counters.

Table 3: Data Columns for Many Spectrometers*

$\text{S}\overset{\text{R}}{+}$	$\text{S}\overset{\text{S}}{+}$	$\text{P}\overset{\text{R}}{+}$	$\text{P}\overset{\text{S}}{+}$
$\text{S}\overset{\text{R}}{-}$	$\text{S}\overset{\text{S}}{-}$	$\text{P}\overset{\text{R}}{-}$	$\text{P}\overset{\text{S}}{-}$
$\text{T}\overset{\text{R}}{+}$	$\text{T}\overset{\text{S}}{+}$	$\text{T}\overset{\text{R}}{-}$	$\text{T}\overset{\text{S}}{-}$

*These spectrometers are Spectrometer H and Ivan in Las Campanas, Spectrometer F in Sutherland, and Spectrometer G in Narrabri. The data-type bitfield contains the following bits: D.STARPORT, D.TWOPOC[†], and D.DELTAB resulting in a decimal value of 98.

[†] Ivan has never had two Pockels cells, however the scalers are already configured for a magnetic Pockels cell and the acquisition software is already configured for one.

DAT files are made up of a series of records. Each line in the file is a record. Under some operating systems (MS-DOS) lines are separated by carriage-return/line-feed pairs. Under other operating systems (Unix) lines are separated by bare line-feeds. Lines may only contain printable ASCII characters.

Table 4: Data Columns for Izaña*

${}_sR$	${}_sS$	${}_TR$	${}_TS$
---------	---------	---------	---------

*The data-type bitfield contains no bits resulting in a decimal value of 0.

Table 5: Data Columns for Mark V in Carnarvon*

${}_sR$	${}_sS$	${}_TR$
---------	---------	---------

*The data-type bitfield contains only the D.LOCKIN bit resulting in a decimal value of 8.

Each record is a space-delimited list of tokens. There are one or more spaces between each token. There may also be one or more spaces at the beginning and end of each line. Records may not be empty—blank lines are not allowed.

Most tokens are ASCII representations of numeric values and can be either integer or floating point. Floating-point values may be in decimal or scientific notation. Some tokens are string representations of dates.

There are two kinds of records: *restart* and *data*. The first record must be a restart record;

Table 6: Data Columns for Jabba in Carnarvon*

$\overset{F}{S}R_+^+$	$\overset{F}{S}S_+^+$	$\overset{F}{P}R_+^+$	$\overset{F}{P}S_+^+$
$\overset{F}{S}R_+^-$	$\overset{F}{S}S_+^-$	$\overset{F}{P}R_+^-$	$\overset{F}{P}S_+^-$
$\overset{F}{S}R_-^+$	$\overset{F}{S}S_-^+$	$\overset{F}{P}R_-^+$	$\overset{F}{P}S_-^+$
$\overset{F}{S}R_-^-$	$\overset{F}{S}S_-^-$	$\overset{F}{P}R_-^-$	$\overset{F}{P}S_-^-$
$\overset{T}{S}R_+^+$	$\overset{T}{S}S_+^+$	$\overset{T}{P}R_+^-$	$\overset{T}{P}S_+^-$
$\overset{A}{S}R_+^+$	$\overset{A}{S}S_+^+$	$\overset{A}{P}R_+^+$	$\overset{A}{P}S_+^+$
$\overset{A}{S}R_+^-$	$\overset{A}{S}S_+^-$	$\overset{A}{P}R_+^+$	$\overset{A}{P}S_+^-$
$\overset{A}{S}R_-^+$	$\overset{A}{S}S_-^+$	$\overset{A}{P}R_-^+$	$\overset{A}{P}S_-^+$
$\overset{A}{S}R_-^-$	$\overset{A}{S}S_-^-$	$\overset{A}{P}R_-^-$	$\overset{A}{P}S_-^-$

*The data-type bitfield contains the following bits: D.TWOPOC, D.STARPORT, D.MAGCAL, and D.FOREAFT resulting in a decimal value of 480.

generally the rest are data records. However, a restart record is written every time data-acquisition begins or when the type of data being collected changes. So if you stop the computer during the day and restart it, another restart record will be written.

The format of the restart record is similar to that of RAWDATA files, except that the number of spectrometers is omitted. While the RAWDATA file contains all of the data collected by all of the spectrometers, at sunset the compacting routine writes out a separate DAT file for each spectrometer. The format of a restart record is:

99.999 *mm-dd-yyyy t* [...]

where *mm-dd-yyyy* is the date and *t* is the data-type bitfield for the data collected.

We have not yet assigned more than sixteen bits, however provision has been made to accommodate extra bits. The highest-order bit (D.MOREBITS) has been reserved as a marker to indicate that another 16-bit bitfield follows.

Data-analysis programs should silently ignore any set bits that are not understood by the program. It is important to be aware that extra bitfields will be present when D.MOREBITS is set.

The remainder of the DAT file is made up of *data* records. There is one record for every forty-second *group*.

The data records contain several fields. The first field is the time (UT) in hours written as a floating-point value. This value will be in the range -12^{h} to $+36^{\text{h}}$ and will correspond to the date in the most recent restart record. All of the following fields are long-integer values. Sums are stored normally, however ratios are multiplied by 10^6 before being written as a long integer. If lock-in amplifiers are used, then the scattered sum is multiplied by 10^8 while the transmitted sum is multiplied by 10^4 .

If the low-speed, magnetic-calibration (delta-B) unit is in use, then the number of data fields is doubled. The unit changes state every four seconds. Alternate records in the raw-data file correspond to the same calibration state. In one forty-second group, there are five records from each state. The compacting routine combines these ten four-second records into one forty-second record, keeping track of the different calibration states separately. When it writes the data to the daily data file, information from both calibration states is written into each data record, thus there are twice as many fields in DAT files as there are in RAWDATA files. All of the fields from one delta-B state are written first. These are followed by all of the fields for the other delta-B state.

The time values in sequential data records (with no intervening restart records) must be (nearly) forty seconds apart. Restart records appear where there are gaps in the data.

4 CMP File Format

In the past, data was sent from the remote sites back to Birmingham on floppy disks. The data was written to these floppy disks as CMP files—compressed, daily data files. The data are no longer transferred using this method, however the CMP format is still being used because the resulting data files are smaller and can be read more quickly by computer. Files can be converted back and forth between the DAT and CMP formats using the **dat-cmp** and **cmp-dat** programs.

CMP files are usually named in a similar manner to DAT files. The name is of the form *synymmdd.cmp* where *s*, *n*, and *yyymmdd* is as for DAT files.

CMP files are copies of DAT files except that they are written in binary form. No actual compression (lossy or non-lossy) is done. In this form, single-precision, floating-point values take four bytes, integers take two bytes, and long integers take four bytes.

CMP files are made up of a series of records. The records are neither fixed in length nor separated by any sort of delimiter. However it is possible to determine the length of the records using the data within the file.

Each records is made up of a series of tokens. The tokens are binary representations of short integers (two bytes), long integers (four bytes), and floating-point values (four bytes). All times are written as single-precision, floating-point values, dates as three short integers, data-type bitfields as short integers, and all data fields as long integers.

There are two kinds of records: *restart* and *data*. The first record must be a restart record; generally the rest are data records. However, a restart record is written every time data-acquisition begins or when the type of data being collected changes. So if you stop the computer during the day and restart it, another restart record will be written.

The format of a restart record is the same as in DAT files. It is very important to observe the value of the data-type field in the restart records. A program that does not understand the data-type field may attempt to read the wrong number of fields and will get hopelessly lost. The number of fields per record is *not* guaranteed to be constant throughout the file. In fact, it is possible to have a data file that contains lock-in amplifier data in the morning and counter data in the evening. Programs that read these files should be aware of this. Also, no information should be assumed based on the origin of the data files. Stations change with time, as do the data files they produce.

Also note that restart records themselves are not guaranteed to be all the same size. This is because there may be more than one data-type field present.

The format of a data record is the same as in DAT files.

5 SM1 File Format

DAT files from Jabba in Carnarvon contain 36 data columns. Because modem time is expensive, we decided to send smaller files back to Birmingham each day. Data in the DAT files were averaged together to produce a smaller file called an SM1 file. Note that this compression is “lossy” — you cannot recover the original DAT file from the SM1 file.

After the computer and software were upgraded [1] in 2002 December, the SM1 files were no longer produced.

SM1 files are actually in the DAT format. The column assignments for these files from Jabba are shown in Table 7.

Table 7: Data Columns in SM1 Files from Jabba*

$\frac{F}{S}R$	$\frac{F}{S}S$	$\frac{T}{R}$	$\frac{T}{S}$	$\frac{A}{S}R$	$\frac{A}{S}S$
----------------	----------------	---------------	---------------	----------------	----------------

*There is only one bit set in the data-type bitfield: D.STARPORT.

6 DATA File Format

A ten-day DATA file is composed of from one to ten days of data. In the past we analyzed our data on an IBM mainframe computer where there was a large overhead for files on disk. Because of this, the ten-day DATA format was created in order to reduce the number of data files.

DATA files have names of the form *ssyynn* or *sssyynn* where *ss* or *sss* is a two- or three-letter code for the station where the data were collected, *yy* is the year without the century, and *nn* is a serial number. The codes for the stations are shown in Table 8. The serial number begins at “01” for the first DATA file of each year and increases by one for each file after that.

There are two methods of choosing which ten days of data are placed in any given DATA file. In the original method, the first ten days of data in each year are placed in the first DATA file for that year. The second ten days of data go in the second DATA file. Note that data are not necessarily collected in all days, hence the actual dates of the first ten days of data are not the same from year to year or even from station to station within the same year.

This method of placing days in the DATA files makes it difficult to locate the data for any specific date. Recently, a new method has been used to choose which days are placed in any given DATA file. The first DATA file of each year (the one with serial number 01) always contains the data from the first ten days of the year, namely January 1–10. The second file contains the data from the second ten days. When data are missing for individual days, the corresponding DATA file contains fewer than ten days of data. In this case there is no “space”

Table 8: Station Codes for DATA Files

<i>Station</i>	<i>Code</i>
Haleakala	HAL
Mount Wilson	MOU
Las Campanas	LAS
Birmingham	BIR
Izaña	TEN
Sutherland	SUT
Carnarvon	OZ
Narrabri	NAR

in the DATA file. This means that, for example, January 10 is not necessarily the tenth day in the file with serial number 01, however, January 11, if data exist for it, will always be in the file with serial number 02.

Within the DATA file, each day begins with a one-line header record, this is followed by the data area. The header record has the form:

mmmmm. gggg ttttt. yymdd. ddd

where *mmmmm* is the modified Julian date, *gggg* is the number of *groups* in the day (see the definition of *group* above on page 2), *ttttt* is the time of the first point in seconds since midnight, *yymdd* is the calendar date, *ddd* is the number of data per point. There are two things to take note of:

1. Certain values are followed by periods. The calendar date is inherently a string and so it is difficult to see how the period could be thought of as a decimal point. The time is a floating-point quantity, however it is always rounded to the nearest second, hence the decimal point seems to be superfluous.
2. Some DATA files contain an additional field following the number of data per point. Its meaning is unknown.

The data area follows the header record and is split into fields. Each field corresponds to a column in the daily data file. The values for each field are contained in the lines following the header record. There are at most six values per line. Each value occupies eleven characters, there are six figures after the decimal point, and there is one additional space after each value. The line is written using the FORTRAN statement:

FORMAT (6(F11.6, 1X))

For Izaña data before 1985 January 1, each value occupied nine characters instead of eleven.

The fields are written consecutively into the DATA file, each field begins on a new line. Hence most of the lines contain six values, however if the number of groups is not divisible by six, the last line of every field will contain less than six values.

The first datum in each field corresponds to the time indicated in the header record. All other data correspond to a time forty seconds later than the immediately preceding datum. This means that data values exist for times when no real data were collected. In these cases, the values are set to zero. Some old DAT files contain data with a cadence of forty-two seconds instead of forty seconds.

There are no blank lines between fields or between days.

7 IZDATA File Format

Before mid 1994 April we received data from Izaña in uncompacted files containing two-second data [2]. They were written on ADFS format disks (Archimedes) with names of the form *ddmmmyy*. These were compacted on an Archimedes computer and written as ASCII files in BBC format on DOS-format disks. The names of these files were of the form *teyymmdd.bbc* or *izyymmdd.bbc*. The BBC format was a hangover from a still earlier processing method when the output files were written by a BBC Master and transferred to a PC by a KERMIT link.

The difference between the *te* and *iz* files is that *iz* files include the transmitted ratio, *te* files do not.

Since mid 1994 April, Izaña data are sent to Birmingham as IZDATA files from Izaña by FTP. They arrive in the */home/stations/izana* directory on *bison.ph.bham.ac.uk*. John Allison copies them to CD-ROMs where they are placed in a directory called *\Izdata*.

IZDATA files have names of the form *izyymmdd.dat*. Notice that the file type is *.dat* but these files are *not* DAT-format files. To complicate things, the data from Izaña for some days is recorded in two or more separate parts. When this occurs, the data files may be named *iayymmdd.dat*, *ibyymmdd.dat*, *icyymmdd.dat*, etc. However, they may be named *izyymmdd.dat*, *ibyymmdd.dat*, *icyymmdd.dat*, etc.

In addition, there are some days for which data are not collected. There will be no IZDATA file for these days.

IZDATA files contain a large number of integer and floating-point numbers, all separated by carriage-return/line-feed pairs. That is, the whole file is made up of lines that each contain one number; there are no spaces or blank lines.

The file begins with an eleven-line header. The eleven numbers in the header are shown in Table 9.

Table 9: IZDATA Header

<i>Line</i>	<i>Item</i>	<i>Range</i>	<i>Type (Units)</i>
1	Day of year	1–366	Integer (days)
2	Day of month	1–31	Integer (days)
3	Month	1–12	Integer (months)
4	Year	1961–2099	Integer (years)
5	Number of Groups	0001–1620	Integer, four digits, zero padded
6	Time check	0–24	Integer (hours UT)
7	Block*	Unknown	Long integer (seconds)
8	Noon [†]	0.0–24.0	floating point (hours UT)
9	Orbital velocity [†]	–1500.0 – +1500.0	floating point (m/s)
10	Spin velocity [†]	–1500.0 – +1500.0	floating point (m/s)
11	Old orbital velocity [†]	–1500.0 – +1500.0	floating point (m/s)

**Block* is the value of the *Clock* at the real time of *Time check*. The *Clock* starts the day at an arbitrary value and is incremented once per second. Thus the header items *Time check* and *Block* allow for the conversion between *Clock* values and real times.

[†]This item is ignored by all analysis software.

The header is followed by a number of eighty-one-line groups. One group is written for every forty seconds of data. The eighty-one numbers in each group are shown in Table 10. The number of groups in the file is indicated in line five of the header.

A dead time correction needs to be applied.

Transmission monitor data from the last two points of each group are ignored. This is because the BBC used to store some house-keeping data there.

The groups in the IZDATA file do not necessarily align with forty-second group boundaries as used by the stations in the rest of the network.

8 BBC File Format

IZDATA files are converted to BBC files by John Allison using the program **readten2**. BBC files are ordinary text files and they have a name of the form *izyymmdd.bbc* or *teyymmdd.bbc* where *yymmdd* is the date on which the data were collected. The difference between the *te* and *iz* files is that *iz* files include the transmitted ratio, *te* files do not.

BBC files may be converted to DAT files using the **bbc-dat** program.

BBC files are made up of a series of data records followed by a four-line trailer. Each line in the file (except the last four) is a record. Under some operating systems (MS-DOS) lines are separated by carriage-return/line-feed pairs. Under other operating systems (Unix) lines are

Table 10: IZDATA groups

<i>Line</i>	<i>Item</i>	<i>Type (Units)</i>
1	<i>dtime</i> *	long integer (seconds)
2	scattered left (1)	long integer
3	scattered right (1)	long integer
4	transmitted left (1)	long integer
5	transmitted right (1)	long integer
6	scattered left (2)	long integer
7	scattered right (2)	long integer
8	transmitted left (2)	long integer
9	transmitted right (2)	long integer
⋮	⋮	⋮
78	scattered left (20)	long integer
79	scattered right (20)	long integer
80	transmitted left (20)	long integer
81	transmitted right (20)	long integer

**dtime* is the value of the *Clock* at the start of the group. You need to use the items *Time check* and *Block* from the header to convert *dtime* to a real time.

separated by bare line-feeds. Lines may only contain printable ASCII characters and must not be more than 256 characters in length, including the carriage return and line feed.

Each data record is a space-delimited list of tokens. There are one or more spaces between each token. There may also be one or more spaces at the beginning and end of each line. Records may not be empty—blank lines are not allowed.

Most tokens are ASCII representations of numeric values and can be either integer or floating point. Floating-point values may be in decimal or scientific notation. Some tokens are string representations of dates.

The bulk of the BBC file is made up of *data* records. There is one record for every forty-second *group*.

The data records contain four or five fields. The first field is the time (UT) in hours written as a floating-point value. This value will be in the range 0^h to +24^h. The time values in sequential data records must be (nearly) forty seconds apart.

The next fields contain the scattered ratio, scattered sum, transmitted sum, and transmitted ratio, in that order. BBC files whose names begin with *te* do not contain the transmitted ratio. The ratios are floating-point values, the sums are long integers.

There are no *restart* records in BBC files.

Table 11: BBC Trailer

<i>Line</i>	<i>String</i>	<i>Header</i>	<i>Range</i>	<i>Type (Units)</i>
1	TOTAL BLOCKS =	5	1–1620	Integer
2	BAD POINTS =	6	1–1620	Integer
3	TIME CHECK	—	0–23	Integer (hours UT)
4	BLOCK	7	Unknown	Long Integer (seconds)

There is a four-line trailer at the end of the file. Each line in the trailer is composed of a descriptive string and a numeric value. There are two spaces before the string and whitespace between the string and the value. Some of the strings contain spaces. The trailer lines are shown in Table 11.

9 SUCIO File Format

When Izaña became a zoo in 2003, it became possible to record time check information for every point. A new data file format was created to include this additional information. We call this format SUCIO after the Spanish word *sucio*, which means “dirty”. This is dirty data.

The names of these files are of the form *izyymmdd.suc* where *yymmdd* is the date. A SUCIO file is made up of a series of records. Each line in the file is a record.

Each record is a space-delimited list of tokens. There are one or more spaces between each token. There may also be one or more spaces at the beginning and end of each line.

Most tokens are ASCII representations of numeric values and can be either integer or floating point. Some tokens are string representations of dates.

There are two kinds of records: *restart* and *data*. The first record must be a restart record; generally the rest are data records. However, a restart record is written every time data-acquisition begins or when the type of data being collected changes. So if you stop the computer during the day and restart it, another restart record will be written.

The format of the restart record is:

```
99.999 mm-dd-yyyy n t1 [...]
```

where *mm-dd-yyyy* is the date, *n* is the number of spectrometers, and *t1* is the data-type bitfield for the data collected by the first spectrometer. Additional data-type bitfields may follow; for example, if more than one spectrometer are in use.

At present, the SUCIO format is only used in Izaña where the BBC scalars record data for only one spectrometer. Thus, *n* is always 1. Also, *t1* is always the only data-type bitfield and its value is always 0.

Table 12: SUCIO Data Records

<i>Field</i>	<i>Item</i>	<i>Type (Units)</i>
1	time	floating point (seconds)
2	BBC clock	integer (seconds)
3	scattered left	long integer (counts)
4	scattered right	long integer (counts)
5	transmitted left	long integer (counts)
6	transmitted right	long integer (counts)

The remainder of the SUCIO file is made up of *data* records. There is one record for every two-second *point*. Each data record contains six fields as shown in Table 12.

time is the time that the data were read from the BBC scalars by the computer. This is approximately two seconds after the start of the point. *time* is written in a floating point format and is the number of seconds since 1970 January 1 00:00 UTC not counting leap seconds. There will always be six digits after the decimal point. It is often more convenient to think of this value as an integer number of seconds, followed by a dot, followed by an integer number of microseconds.

BBC clock is the value read from the clock in the BBC scalars. This clock is a ten-digit counter that increments by two for each two-second point. This counter thus returns the time in seconds since an arbitrary epoch. The BBC clock drifts very slowly with respect to UTC.

The remaining four fields in the SUCIO file are the four values read from the data counters. No dead-time corrections have been applied to these numbers.

10 RES File Format

Daily data files are processed by ORANGE [3] to produce residual velocity data. These data are written to RES files. Older RES files normally have a name of the form *snymmdd.res* where *s* is the first letter of the station name; *n* is the second letter of the station name, or “b” for the second instrument at a station; and *ymmdd* is the date on which the data were collected.

There is now a new RES-file-naming scheme. RES files now have a name of the form *snymmdd-qualifiers.res* where *snymmdd* is as before and *qualifiers* is a list of qualifier/value pairs. There are one or more pairs in each name. Pairs are concatenated together (there is no separating space, comma, or other character). Each qualifier is an uppercase letter. Each value is one or more lowercase letters. We have defined six qualifiers:

D	detector
M	magnet
B	magnetic

F filter
S selection
O other

The qualifiers should appear in the order specified in the table above. D specifies which detectors were used and may take one of four values:

Ds starboard
Dp port
Dm mean of starboard and port
Dd difference between starboard and port

The D qualifier is required; the others are optional. M specifies which magnet was used and may take one of four values:

Mf fore
Ma aft
Mm mean of fore and aft
Md difference between fore and aft

When M is omitted, Mf is assumed. B indicates that the data are magnetic residuals as opposed to velocity residuals and may take one of five values:

Bs velocity-calibrated, single-fit magnetic residuals
Bd velocity-calibrated, double-fit magnetic residuals
Bb magnetic-calibrated, blue-wing magnetic residuals
Br magnetic-calibrated, red-wing magnetic residuals
Bm magnetic-calibrated, mean of red- and blue-wing magnetic residuals

F specifies any filters [4] that have been used and may take the following values:

Ff multi-parameter footprint correction
Fm 25-point moving-mean high-pass filter
Fs sync high-pass filter

The F qualifier may take more than one value. For example, Ffm indicates that both footprint correction and moving-mean filtering has been done. The order of the values for the F qualifier indicate the order in which the filters were applied.

S specifies if the residuals have been optimized for a particular bandwidth and may take one of the following values:

Sg	Optimized for low-frequency <i>g</i> -mode regime (200–700 μ Hz)
Sp	Optimized for low-frequency <i>p</i> -mode regime (800–1,300 μ Hz)
Sh	Optimized for high-frequency regime (5,500–12,500 μ Hz)

If S is omitted then no optimization has taken place. O is used for anything other than that specified above and may take one or more of the following values:

Od	Delta-B residuals
Ob	Blue wing only
Or	Red wing only

If more than one value is specified, the values should appear in the same order as the table above. No qualifier may appear more than once. Ffm is ok, but FfFm is not.

As an example, residuals from Carnarvon on 2004 June 21 that have been processed with a footprint correction and a moving-mean filter should be in a file called *ca030621-DmFfm.res*.

RES files are made up of a series of records. Each line in the file is a record. Under some operating systems (MS-DOS) lines are separated by carriage-return/line-feed pairs. Under other operating systems (Unix) lines are separated by bare line-feeds. Lines may only contain printable ASCII characters

Each record is a space-delimited list of tokens. There are one or more spaces between each token. There may also be one or more spaces at the beginning and end of each line. Records may not be empty—blank lines are not allowed.

Tokens are ASCII representations of numeric values and can be either integer or floating point. Floating-point values may be in decimal or scientific notation. Some tokens are string representations of dates.

There are two kinds of records: *restart* and *data*. The first record must be a restart record; generally the rest are data records.

The format of a restart record is:

```
99.999 mm-dd-yyyy t [...]
```

Table 13: RES Data-Type Bitfield 1

<i>Bit</i>	<i>Constant</i>	<i>Data Collected</i>
0	RES_MMEAN	Residuals were filtered with a moving mean.
1	RES_STARBOARD	Data collected by starboard detector only.
2	RES_PORT	Data collected by port detector only.*
3	RES_AFT	Data collected by aft cell.
4	RES_MARK_I	Data collected by Mark I in Izaña.
5	RES_MARK_IV_H	Data collected by Mark IV in Haleakala.
6	RES_MARK_IV_M	Data collected by Mark IV in Mount Wilson.
7	RES_MARK_V	Data collected by Mark V in Carnarvon.
8	RES_SPEC_F	Data collected by Spectrometer F in Sutherland.
9	RES_SPEC_G_B	Data collected by Spectrometer G in Birmingham.
10	RES_SPEC_G	Data collected by Spectrometer G in Narrabri.
11	RES_SPEC_H	Data collected by Spectrometer H in Las Campanas.
12	RES_IVAN	Data collected by Ivan in Las Campanas.
13	RES_JABBA	Data collected by Jabba in Carnarvon.
14	RES_KLAUS	Data collected by Klaus in Mount Wilson.
15	RES_MOREBITS	More bitfields follow.

*If both RES_STARBOARD and RES_PORT are present, the file contains residuals of the difference between the two detectors. If neither bit is present, the the mean of the two detectors has been used.

where *mm-dd-yyyy* is the date that the data were collected and *t* is the data-type bitfield for the data collected.

At some stations, data are collected beyond 24^h UT, which means that the date changes during data acquisition. At these stations, a special method is used to determine the date to write in the restart records. The dates in the restart records always corresponds to the times in the data records and the combination of the two are chosen so that there is no date change or discontinuity in the time value at any time during the day. At some stations, data files will contain times greater than 24^h UT, while at others the data files will contain times less than 0^h UT.

The data-type field is a bitfield—each bit has a different meaning. Symbolic constants are defined in the software to help you work with data types. Tables 13 and 14 summarize the data-type bitfields. The highest-order bit (RES_MOREBITS) in each bitfield is a marker to indicate that another 16-bit bitfield follows.

Data-analysis programs should silently ignore any set bits that are not understood by the program. It is important to be aware that extra bitfields will be present when RES_MOREBITS is set.

If all three bits RES_NPOLY0, RES_NPOLY1, and RES_NPOLY2 are zero, the fit contains three coefficients. This is because early versions (v1.05 and earlier) of ORANGE did not put these bits into the residual files. A change in the value of NPOLY from 3 to 4 was made in ORANGE v1.6.

Table 14: RES Data-Type Bitfield 2

<i>Bit</i>	<i>Constant</i>	<i>Data Collected</i>
0	RES_NPOLY0	} The number coefficients in the fit. This is a three-bit value from 0 to 7.
1	RES_NPOLY1	
2	RES_NPOLY2	
3	RES_AFT2	Modifies RES_AFT.*
4	RES_FOOTPRINT	Footprint correction applied.
5	RES_SYNC	Sync high-pass filter applied.
6	RES_SELECTG	Optimized for <i>g</i> -modes (200–700 μ Hz).
7	RES_SELECTP	Optimized for <i>p</i> -modes (800–1, 300 μ Hz).
8	RES_SELECTH	Optimized for 5,500–12, 500 μ Hz.
9	RES_DELTAB	Delta-B residuals.
10	RES_MAGNETIC	Magnetic residuals.
11	RES_BLUE	Blue wing only.
12	RES_RED	Red wing only.
15	RES_MOREBITS	More bitfields follow.

*RES_AFT2 modifies the meaning of RES_AFT. Both are present when the mean of fore and aft are being used. RES_AFT2 is present without RES_AFT when the difference is being used.

The remainder of the DAT file is made up of *data* records. There is one record for every forty-second *group*.

The data records contain two fields. The first field is the time (UT) in hours written as a floating-point value. This value will be in the range -12^{h} to $+36^{\text{h}}$ and will correspond to the date in the most recent restart record. The second field is the residual velocity in meters per second and is also written as a floating-point value.

References

- [1] BREK A. MILLER. The grand opening of the Carnarvon Zoo in 2002 November. *BISON Technical Report Series*, Number 193, High-Resolution Optical-Spectroscopy Group, Birmingham, United Kingdom, January 2003.
- [2] RACHEL HOWE. Instructions for use of ALLFIT. *BISON Technical Report Series*, Number 36, High-Resolution Optical-Spectroscopy Group, Birmingham, United Kingdom, June 1995.
- [3] BREK A. MILLER. RED v1.6—A cubic fit. *BISON Technical Report Series*, Number 198, High-Resolution Optical-Spectroscopy Group, Birmingham, United Kingdom, February 2003.

- [4] WILLIAM J. CHAPLIN, YVONNE P. ELSWORTH, GEORGE R. ISAAK, KONSTANTIN I. MARCHENKOV, BREK A. MILLER, ROGER NEW, BALAZS PINTER, AND THIERRY APPOURCHAUX. Peak finding at low s/n: low- ℓ solar acoustic eigenmodes at $n \leq 9$ from the analysis of BiSON data. *Monthly Notices of the Royal Astronomical Society*, **336**, 979–991, 2002.