

## STATION PARAMETERS

The station parameters provide a small amount of information relating to the field station where the data is being recorded and the equipment used. The parameters were designed “in-house” to be contained in a one-line file **stn.dat** or as a one-line header in the data files. The parameters are defined below and are written to file using the following [format string](#):

```
'(i5,x,2(f7.2,x),a20,x,d16.12,x,2(f10.7,""),f10.7,x,a1,x,a)'
```

### 1. Station number

This parameter is an identifier number for the station and is either the five digit IUWDS code (as described in the International URSIgram and World Days Service Synoptic Codes for Solar and Geophysical Data) or the four digit SWS station code.

### 2. Latitude and Longitude

These two parameters are the station geographic latitude and longitude and are in decimal degrees with negative values for south and west respectively.

### 3. Components

This parameter is a string indicating the geomagnetic field components being measured. The components are separated by commas and correspond to the order the data is recorded in the files. The geomagnetic field data are usually provided for the geographic north-south, east-west and vertical directions (X,Y,Z) or geomagnetic north-south, east-west and vertical directions (H,D,Z). Upper case letters indicate the data being measured is total field data and lower case letters indicate the data being measured is “variometer” data (the variations of the Earth’s magnetic field). Sometimes other letters in the components string are used to indicate the organisation that owns the magnetometer sensor:

- GA - [Geoscience Australia](#)
- SERC - [International Center for Space Weather Science and Education](#)

### 4. Sample Frequency

This parameter is the sample frequency (in seconds) used to record the data and applies to all the geomagnetic field components.

### 5. Channel Resolution

These three parameters (separated by commas) are the resolution of the data logging system A/D converter in nT/LSB (nanoTesla/Least Significant Bit) for each of the geomagnetic field components (3) being recorded.

### 6. Identifier Letter

This parameter is a letter used to identify the type of data being recorded according to the following list:

“a”: Total field data.

“b”: Variometer data with no offset and one resolution parameter for all components.

“c”: Variometer data with an offset and one resolution parameter for all components.

“d”: Data of type “b” with a change made to the resolution parameter.

“e”: Data of type “b” with separate resolution parameters for each component which may have changed from the original single parameter resolution value.

“f”: Data of type “c” with separate resolution parameters for each component which may have changed from the original single parameter resolution value.

“g”: Data of type “f” with a change made to one or all of the resolution parameters.

“h”: Data of type “e” with a change made to one or all of the resolution parameters.

“i”: Data recorded using IPSMAG 2.x data acquisition system.

“j”: Data recorded using SWNMAG 1.x data acquisition system.

“o”: Data from a [Geoscience Australia](#) standard geomagnetic observatory.

“p”: Geomagnetic pulsation data recorded from Induction magnetometers owned by the [University of Newcastle](#).

“t”: Data used for test purposes only.

“G”: Preliminary IAGA-2002 format data from [Geoscience Australia](#) near real-time observatories.

“I”: Total field data recorded using IPSMAG 2.x data acquisition system.

“J”: Total field data recorded using SWNMAG 1.x data acquisition system.

## 7. Data Format String

This parameter is the format string that has been used to write the data to file and uses the FORTRAN formatting syntax. The format string has the form:

```
FORMAT = '(q1f1s1f2s2 ... fnqn)'
```

where q, f, and s are described below.

### Record Terminators

q is zero or more slash (/) record terminators. On output, each record terminator causes the output to move to a new line.

### Format Codes

*f* is a format code. Some format codes specify how data should be transferred while others control some other function related to how output is handled. The code *f* can also be a nested format specification enclosed in parentheses. This is called a *group specification* and has the following form:

...[*n*](*q1f1s1f2s2 ... fnqn*) ...

A group specification consists of an optional repeat count *n* followed by a format specification enclosed in parentheses.

### “A” Format Code

The A format code transfers character data. The format is

[*n*]A[*w*]

where:

*n* — is an optional repeat count ( $1 \leq n$ ) specifying the number of times the format code should be processed. If *n* is not specified, a repeat count of one is used.

*w* — is an optional width ( $1 \leq w$ ) specifying the number of characters to be transferred. If *w* is not specified, the entire string is transferred. If *w* is greater than the length of the string, the string is right justified.

### “:” Format Code

The colon format code terminates format processing if there are no more data remaining in the argument list.

### “\$” Format Code

If a “\$” format code is found in the format specification, this default newline character is not output.

### “F,” “D,” “E,” and “G” Format Codes

The F, D, E, and G format codes are used to transfer floating-point values to the file. The format is

[*n*]F[*w.d*]

[*n*]D[*w.d*]

[*n*]E[*w.d*] or [*n*]E[*w.dEe*]

[*n*]G[*w.d*] or [*n*]G[*w.dEe*]

where

*n* — is an optional repeat count ( $1 \leq n$ ) specifying the number of times the format code should be processed. If *n* is not specified, a repeat count of 1 is used.

*w.d* — is an optional width specification ( $1 \leq w \leq 256$ ,  $1 \leq d < w$ ). The variable *w* specifies the number of characters in the external field. For the F, D, and E format codes, *d* specifies the number of positions after the decimal point. For the G format code, *d* specifies the number of significant digits displayed.

*e* — is an optional width ( $1 \leq e \leq 256$ ) specifying the width of exponent part of the field.

The F and D format codes are used to output values using fixed-point notation. The value is rounded to *d* decimal positions and right-justified into an external field that is *w* characters wide. The value of *w* must

be large enough to include a minus sign when necessary, at least one digit to the left of the decimal point, the decimal point, and  $d$  digits to the right of the decimal point. The E format code is used for scientific (exponential) notation. The value is rounded to  $d$  decimal positions and right-justified into an external field that is  $w$  characters wide. The value of  $w$  must be large enough to include a minus sign when necessary, at least one digit to the left of the decimal point, the decimal point,  $d$  digits to the right of the decimal point, a plus or minus sign for the exponent, the character “e” or “E”, and at least two characters for the exponent. The G format code uses the F output style when reasonable and E for other values, but displays exactly  $d$  significant digits rather than  $d$  digits following the decimal point. If the field provided is not wide enough, it is filled with asterisks (\*) to indicate the overflow condition.

### “I,” “O,” and “Z” Format Codes

The I, O, and Z format codes are used to transfer integer values to the file. The I format code is used to output decimal values, O is used for octal values, and Z is used for hexadecimal values.

The format is as follows:

$[n]I[w]$  or  $[n]I[w.m]$

$[n]O[w]$  or  $[n]O[w.m]$

$[n]Z[w]$  or  $[n]Z[w.m]$

where

$n$  — is an optional repeat count ( $1 \leq n$ ) specifying the number of times the format code should be processed. If  $n$  is not specified, a repeat count of 1 is used.

$w$  — is an optional integer value ( $1 \leq w \leq 256$ ) specifying the width of the field in characters. If the field provided is not wide enough, it is filled with asterisks (\*) to indicate the overflow condition.

$m$  — specifies the minimum number of nonblank digits required ( $1 \leq m \leq 256$ ). The field is zero-filled on the left if necessary. If  $m$  is omitted or zero, the external field is blank filled.

### Quoted String and “H” Format Codes

Any quoted strings or Hollerith constants are sent directly to the output. The format for a Hollerith constant is:

$nHc1c2 c3 \dots cn$

where

$n$  — is the number of characters in the constant ( $1 \leq n \leq 255$ ).

$ci$  — is the characters that make up the constant. The number of characters must agree with the value provided for  $n$ .

### “T” Format Code

The T format code specifies the absolute position in the current record. The format is:

$Tn$

where

$n$  — is the absolute character position within the record to which the current position should be set ( $1 \leq n$ ).

T — differs from the TL, TR, and X format codes primarily in that it requires an absolute position rather than an offset from the current position.

#### **“TL” Format Code**

The TL format code moves the current position in the external record to the left. The format is:

$TLn$

where

$n$  — is the number of characters to move left from the current position ( $1 \leq n$ ). If the value of  $n$  is greater than the current position, the current position is moved to column one.

TL — is used to move backwards in the current record.

#### **“TR” and “X” Format Codes**

The TR and X format codes move the current position to the right. The format is:

$TRn$

$nX$

where

$n$  — is the number of characters to skip ( $1 \leq n$ ). The current output position is moved  $n$  characters to the right.

The TR or X format codes can be used to leave whitespace in the output. These two format codes differ in one way. Using the X format code at the end of an output record will not cause any characters to be written unless it is followed by another format code that causes characters to be output. The TR format code always writes characters in this situation.

#### **Field Separators**

s is a field separator. A field separator consists of one or more commas (,) and/or slash record terminators (/). The only restriction is that two commas cannot occur side-by-side.