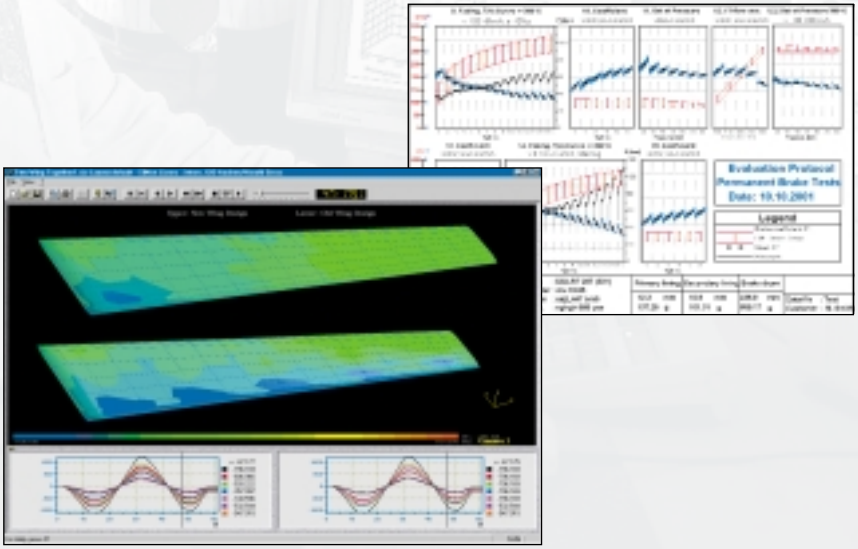
DIAdem™

Data Set Description



ni.com/diadem

NATIONAL INSTRUMENTS™

DIAdem™

DIAdem Data Set Description Manual

NATIONAL INSTRUMENTS™

Februar 2002 Edition

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 001 512 683 0100

Worldwide Offices

Australia 03 9879 5166, Austria 0662 45 79 90 0, Belgium 02 757 00 20, Brazil 011 284 5011,
Canada (Calgary) 403 274 9391, Canada (Ontario) 905 785 0085, Canada (Québec) 514 694 8521,
China 0755 3904939, Denmark 45 76 26 00, Finland 09 725 725 11, France 01 48 14 24 24,
Germany 089 741 31 30, Greece 30 1 42 96 427, Hong Kong 2645 3186, India 91805275406,
Israel 03 6120092, Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico (D.F.) 5 280 7625,
Mexico (Monterrey) 8 357 7695, Netherlands 0348 433466, Norway 32 27 73 00, Poland 48 22 528 94 06,
Portugal 351 1 726 9011, Singapore 2265886, Spain 91 640 0085, Sweden 08 587 895 00,
Switzerland 056 200 51 51, Taiwan 02 2377 1200, United Kingdom 01635 523545

For further support information, see the *Technical Support* appendix. To comment on the documentation, send e-mail to techpubs@ni.com

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

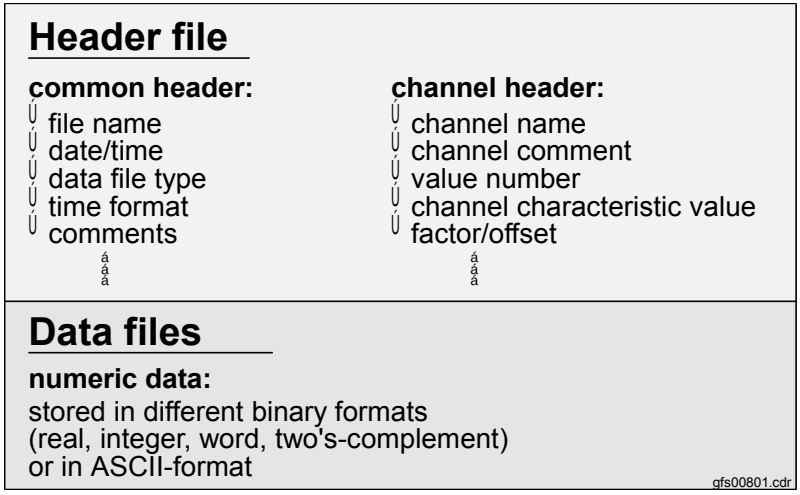
1 Data Organization in DIAdem

This booklet describes the structure of the DIAdem header file.

However, for better understanding, we start with the principles of the data organization in DIAdem. This information is useful for the customer who would like to access DIAdem files with his own programs, or who would like to write DIAdem compatible files.

DIAdem uses a file construct which can be adapted to match different file formats to the maximum extent. This is achieved by storing the **data** itself and the **organizational information** required for administering the data in different files.

Internal organization of the data in DIAdem



The administration file with the organizational information in DIAdem is called the header file; the file(s) with the actual data are called the data files. Both file-types together form a data set (see the diagram above).

1.1 Header File

The **header file** contains all the information that is required for loading the (numeric) data files. The header file contains two kinds of information. Firstly, there is a general description of the data file, the ***general header information***. Secondly, all the information which is required for reading the individual channels of the data

file(s), the ***channel header information***, is stored in the header file.

In addition, the **channel headers** can hold certain information about certain parameters of the individual channels (e.g. minimum value, monotony). The **file extension** of the header files is always DAT and is assigned mandatorily.

1.2 Data File

The **data files** (one or even more per data set) contain the numeric data. One or more channels can be stored in every data file. In all, DIAdem can read and process more than 65,000 channels with about 2 billion values.

For the data files, DIAdem supports the binary formats listed below and the ASCII format:

FORMAT	MEANING
REAL32	Data in 4-byte real format (32 bit)
REAL48	Data in 6-byte real format (48 bit)
REAL64	Data in 8-byte real format (64 bit)
MSREAL32	Data in 4-byte real format (32 bit) [1])
INT16	Data in 2-byte integer format (16 bit) [4])
INT32	Data in 4-byte integer format (32 bit) [4])
WORD8	Data in 1-byte Word format (8 bit) [5])
WORD16	Data in 2-byte Word format (16 bit) [5])
WORD32	Data in 4-byte Word format (32 bit) [5])
TWOC12	Data in 12-bit integer format [2]) [4])
TWOC16	Data in 16-bit integer format [2]) [4])
ASCII	ASCII[3])

[1]) The data format ***MSREAL32*** is a 4-byte real format which was used in earlier versions of Microsoft programming languages.

[2]) The data formats ***TWOC12*** and ***TWOC16*** are special formats (two-complement), which are written directly to the hard disk by analog-digital converter cards.
[3]) In the ***ASCII*** format, the numeric data is stored as an *ASCII*-character string in an *ASCII*-text file.
[4]) signed
[5]) unsigned

2 Structure of the DIAdem Header File

The structure of the header file of DIAdem is described below. The theoretical structure of the header file is followed by the detailed documentation of the *general header* and the *channel header*.

The following **diagram** contains a brief description of the structure and the contents of the header file.

Structure of the DIAdem header file

Entry in the header file	Meaning
DIAEXTENDED {@:ENGLISH	**Type of data record:** Keyword for DIAdem-data files
#BEGINGLOBALHEADER *general header entries* *#ENDGLOBALHEADER*	**General data set description:** contains entries regarding the structure of the entire data file
#BEGINCHANNELHEADER *header entries for first channel* *#ENDCHANNELHEADER* *#BEGINCHANNELHEADER* *Header entries for 2nd channel* *#ENDCHANNELHEADER* • • *#BEGINCHANNELHEADER* *header entries for nth channel* *#ENDCHANNELHEADER*	**Description of the channels of the data set:** contains entries for the structure of all the channels which are stored in the data file

The left part of the **diagram** shows the general structure of the header file, consisting of the *data set type,* the *general header* and the *channel headers*.

The header files must always start with the keyword DIAEXTENDED {@:ENGLISH , which specifies the *data set* type.

The individual header blocks are also started and terminated with keywords. The header entries are stored within these blocks.

All header entries are mandatorily started with a key number (an identification number) **and a comma.**

```
101,Engine characteristic line diagram
```

Lines which do not start with a number can be used as **comment lines**. Hence, there can be any number of blank lines present in the header file.

3 General Data Header

The *general header* contains information about the entire data file. Apart from general comments on the contents of the file, the person processing it and the file description, it is also possible to store, for example, the time format, the value for NoValues, as well as the date and time when the file was last saved.

The following **diagram** shows an overview of all the possible entries in the general header. All the *general header file entries* are explained in detail right after the diagram.

Possible entries in the general header

GENERAL HEADER	
DIAEXTENDED {@:ENGLISH	Keyword for the type of data set
#BEGINGLOBALHEADER	Keyword for the start of the general header
1 : Keyword for the origin of the data set 2 : Revision number 101: Description of the data set 102: Comments on the data set 103: Person processing the data set 104: Date 105: Time 106: Description of the comments 110: Time format for time-channels in the case of ASCII-files 111: Value for NoValues in the data file 112: Interchange high- and low-bytes 130: Reserve 1 131: Reserve 2 132: Reserve 3 133: Reserve 4	
#ENDGLOBALHEADER	Keyword for the end of the general header

3.1 Syntax of the Header Entries

Basically, only the header data which is required for describing the data file is stored in the header file. Entries which are not needed in the global header can be simply omitted.

The header entries always start with an identification number, followed by a comma. The actual entries for the header data follow immediately thereafter. For example,

```
101,Engine characteristic line diagram
```

is a valid entry for the description of the data set.

3.2 Explanation of the Header Entries

The following text gives detailed information on the possible entries in the ***general header***. If required, all the keywords are also explained.

The identification numbers which have been listed in the diagram above have been used for documenting the individual header entries. This will facilitate a simple correlation of the detailed description to the identification numbers in the overview diagram.

1: Keyword for the origin of the data set
An identifier is entered in the data file, giving information about the operating system under which the data set has been stored.

```
1,WINDOWS 32Bit
```

2: Keyword for the Revision of DIAdem
This keyword provides information about the revision, build and filter release number of DIAdem under which the data set has been stored.

```
2,{@R:800 {@V:8.00.981 {@F:4.00
```

101: Description of the data set
General comment on the description of the data.

```
101,Engine characteristic line diagram
```

102: Comments on the data set (Array [1..100])
A total of 20 lines for commenting the entire data file, can be entered in the general comment field.

```
102,3D data for calculating an engine
characteristic line diagram
```

103: Person processing the file
Name of the person or any other comment.

```
103,Peter Smith
```

104: Date
Date on which the file was last saved. This entry is automatically updated when the data file is saved in DIAdem.

```
104,02.20.2002
```

105: Time
Time when the file was last saved. This entry is automatically updated when the data file is saved in DIAdem.

```
105,10:28:26
```

106: Description of the comments (Array [1...100])
You can write a maximum of 100 comments in the data set properties for the current stock of data. Each comment can also be given its own designation, allowing more flexible access to the data set comment, for example, in Autosequences or when designing graphics.

```
106,A comment
```

The designations and the comments have to be in the same order.

110: Time format for time channels in the case of ASCII-files
This header entry is only required for ASCII data files, and has no significance for binary files. The ***time format***-specification defines in which format the time-data is stored in the ASCII-file to be read. A time format specification is generally started with the # character, followed immediately by the actual format specification.

```
110,#mm.dd.yyyy hh:nn:ss
```

Leading and trailing spaces in the format specification are removed when the file is read and then not considered any more.

The following time formats are available in DIAdem :

'm'	Place-marker for month
'd'	Place-marker for day
'y'	Place-marker for year
'h'	Place-marker for hours
'n'	Place-marker for minutes
's'	Place-marker for seconds

If the year is only given in two digits, 2000 must be added to it, so that DIAdem reads in the correct year.

01.01.02 :# dd.mm.yy+2000

111: Value of NoValues in the data file
Specification of the numerical value which is to be interpreted as ***NoValue*** for floating point values in the entire data file. *NoValues* are undefined values of a channel, e.g. missing readings in a series of measurements, which could not be determined for some reason or other.

```
111,9.90000000000000E+0034
```

In the *channel headers* (see entry 254) a NoValue can also be additionally specified for each individual channel. If this parameter is not specified in the general header and the channel headers, DIAdem assumes a standard default value of **9.9E+34** as the NoValue.

112: Exchange high and low bytes
This entry specifies whether the data in the binary file being read is to be read from highbyte to lowbyte or vice versa.

```
112,High -> Low
```

- Byte sequence: *High -> Low*
  Computers with processors of type 8086, 80x86 (Intel, IBM-PC and compatibles) use this byte sequence.
- Byte sequence: *Low -> High*
  Binary files on computers with processors of type 680x0 (Motorola, e.g. HP9000) exist in this format.

If this header entry is not given, DIAdem assumes the following settings:

4 Channel Headers

Every header file in DIAdem contains a general header and one or more channel headers. The number of channel headers in the header file depends on the number of channels stored in a data set. Theoretically, up to 65,000 channel headers and hence the same number of channels can be administered by DIAdem.

The channel headers contain information on the individual channels of the data file(s). In addition to comments on the channels (name, comments, unit), other channel characteristics (minimum, monotony etc.), the number of values as well as parameters for reading the channels can be stored there.

The following diagram gives an overview of all the possible entries in a channel header.

CHANNEL HEADERS (CHANNEL 1)	
#BEGINCHANNELHEADER	Keyword for the start of the first channel header
200 :	Channel name
201 :	Channel comment
202 :	Unit
210 :	Channel type
211 :	File from which channel data is read
212 :	Used internally
213 :	Method of storing the data
214 :	Data type
215 :	Bit masking
220 :	No. of values in the channel
221 :	Pointer to the 1st value in the channel
222 :	Offset for ASCII block files Offset for binary block files with header
223 :	Local ASCII-pointer in the case of ASCII block files
230 :	Separator character for ASCII-block files
231 :	Decimal character in ASCII-files
232 :	Exponential character in ASCII-files
240 :	Starting value / Offset
241 :	Step width / Factor
242 :	Used internally
250 :	Minimum value in the channel
251 :	Maximum value of the channel
252 :	Keyword for NoValues in the channel
253 :	Keyword for monotony
254 :	Value for NoValues in the channel
260 :	Keyword for the data display at the interface
270 :	Register variable RV1
...	to
275 :	RV6 for storing the channel-related additional data
280 :	Register variable Int1

CHANNEL HEADERS (CHANNEL 1)	
... to 284 : Int5 for storing the channel-related additional data in Integer format 300 : Reserve 1 301 : Reserve 2	
#ENDCHANNELHEADER	Keyword for the end of the first channel header

4.1 Syntax of the Channel Header Entries

Basically, only header data which is required for describing the data file is stored in the header. Entries which are not required in the global header can be simply omitted.

The header entries always start with an identification number followed by a comma. The actual entries for the header data follow immediately thereafter. As an example,

```
200,time-channel
```

is a valid entry for the designation of the data set.

4.2 Explanation of the Channel Header Entries

Detailed information on the possible entries in the ***channel headers*** is given below. If required, all the keywords have also been given.

The same identification numbers listed in the diagram above have been used in documenting the individual header entries This will facilitate correlation of the detailed description with the identification numbers in the overview diagram.

200: Name of the data channel (channel name)
Channel name for the designation of the data channel.

```
200,time-channel
```

201: Comments on the channel data
Comments on the respective channel contents.

```
201,driving time
```

202: Designation of unit
Unit of the data stored in the channel.

```
202,s
```

210: Channel type
The channel type is a keyword which provides information on the method of storing the data.

In DIAdem, channels can be defined implicitly and explicitly. ***Implicit*** channels are defined by their starting value and the step between two consecutive values (step-width). ***Explicit*** channels exist value for value. If required, a factor and an offset can be given for converting explicit channels into the corresponding physical quantities.

```
210,EXPLICIT
```

211: Name of the data file with the channel data
Name of the data file (with extension), in which the numeric data of the channel to be read is stored.

```
211,DRIVE.R64
```

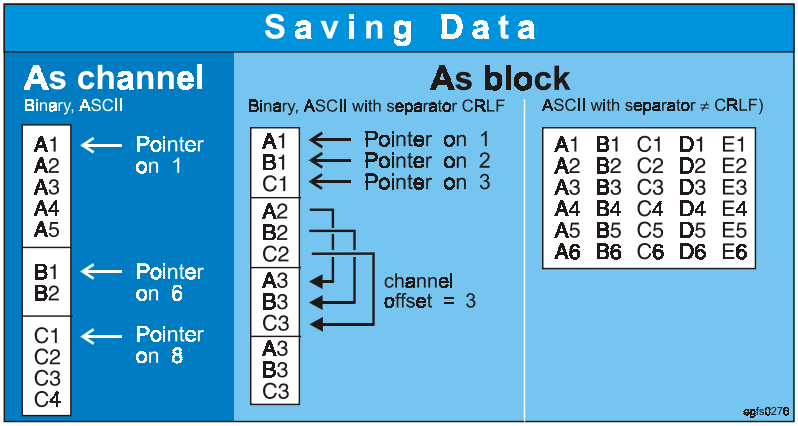
The filename extension must be specified and the data file must be located on the configured data path, where the header file is located.

213: Method of storing the data
The method of storing the data indicates whether the data is stored in a channel-oriented or a block-oriented manner. The possible values are CHANNEL and BLOCK.

```
213,BLOCK
```

In the case of ***implicit*** data channels, the entry for the data storage method is not required, since this data is generated. Any entries made are ignored.

Block-/channel-oriented storage of the data



If a data file is constructed **channel-wise** (columns of the matrix), all the channel data of the first channel is first stored in succession, then that of the second channel, and so on (see **diagram**). In this method of storage, the channel lengths of the individual file channels can be different.

In the case of **block-wise** storage (rows of the matrix), in contrast, the first values of all the data channels are stored first in the data file, then the second values of all data channels, and so on (see **diagram**). This method of storage is applied, for example, when, in a time-dependent measurement, the readings of all measurement channels are stored immediately after every time-step.

In the case of files which exist in the ASCII block format, the individual values are stored in **one** line, delimited by **separators** (e.g. *commas* ","; cf. **diagram**). The number of channels to be stored or read in this manner is limited to 255. The length of the ASCII-line is optional. This does not apply to ASCII-block files with the separator CRLF (see. **diagram**), for which the number of channels is not limited.

In the case of block-wise storage, variable lengths of the data stored in the data channels are not permissible.

214: Data type

The data format of the data to be read is defined by the data type. The DIAdem supported formats for reading are described briefly below.

```
214,ASCII
```

KEYWORD	SUPPORTED FORMAT
REAL32	Data in 4-byte real format (32 bit)
REAL48	Data in 6-byte real format (48 bit)
REAL64	Data in 8-byte real format (64 bit)
MSREAL32	Data in 4-byte real format (32 bit)
INT16	Data in 2-byte integer format (16 bit)
INT32	Data in 4-byte integer format (32 bit)
WORD8	Data in 1-byte Word format (8 bit)
WORD16	Data in 2-byte Word format (16 bit)
WORD32	Data in 4-byte Word format (32 bit)
TWOC12	Data in 12-bit integer format
TWOC16	Data in 16-bit integer format
ASCII	ASCII

The data formats TWOC12 and TWOC16 are special formats (Two's complement format), which are written directly to the hard disk by AD-converter cards.) Data in the 4-byte Real format (32 bit): [MSREAL32]. The data format MSREAL32 is used by earlier Microsoft-programs.

215: Bit masking

Bitmasking makes it possible to **hide bits during the read operation** (to mask them). This entry is a program-internal entry of type *LongInt* (4 byte sign-prefixed) and can therefore mask data up to a size of 4 bytes. The masking is done with the help of an AND-linkage:

New value = old value AND bit mask

As an example, to mask the 5th bit from a value, the bit mask should be set to $2^{5-1} = 16$. If the 5th bit in the field is set, the result is *16*, else it is *0*. On reading the file into DIAdem, therefore, there results a channel which contains only the values *0* and *16*.

However, if the channel is to have only the values 0 (*bit not set*) and 1 (*bit set*), the channel should be scaled; for the example above, the factor (channel header entry[241]) should be set to $^1/_{16} = 0.0625$.

If **several individual bits** are to be read separately from one value, a separator channel header should be generated for every bit to be masked out. For example, if the 1st and 7th bit is to be read and scaled from a value, then two channel headers with the following entries should be generated.

Channel header 1 (1st bit):	*215,1* *241,1*
Channel header 2 (7th bit):	*215,64* *241,0.01562* (=1/64)

Of course, it is also possible to **mask several bits** and save the result in a channel. E.g. if the 3rd and 8th bits are to be masked, the value

$$2^{3-1} + 2^{8-1} = 132$$

should be specified for the bit-masking. Then, on reading the file in DIAdem , there results a channel which contains only the following values:

0	: Bit 3 and bit 8 not set
4	: Bit 3 set, bit 8 not set
128	: Bit 3 not set, bit 8 set
132	: Bit 3 and bit 8 set

220: No. of values (channel length)
The no. of values specifies the number of values to be read into the channel. In the case of data which is stored channel-wise, this value can be smaller than the actual number of data items stored in the file.

```
220,1024
```

In the case of **binary files which are stored block-wise,** the specified number of values must tally exactly with the number of values existing in the file to be read. If that is not the case, the offset (see 'channel header entry [222]) must be defined explicitly.

221: File offset: Pointer to the first date of a channel
The pointer to the first date of a channel specifies the position within a data file at which the first value of the desired channel is located. In the case of data in **ASCII-format a line number is expected here;** in the case of data in **binary format, a record number.**

```
221,1
```

The file offset should be specified for the following data files:

- ❍ data stored channel-wise
- ❍ data stored block-wise, with header
- ❍ ASCII block files with separator CR/LF:

The first value of a channel is located at the position specified by the file-offset. Starting from the starting point to be specified in this field, the channel values are read from the data file. The second value of block-wise stored data is located at *file offset + channel number,* the third value at the position *file offset + (2 · channel number) etc.* The channel number (channel offset) here should be specified separately in a special entry (see channel header entry [222]).

The file offset is not required only in the case of ASCII block files (separator not CR/LF), and binary files created block-wise, which do not have any header.

❍ Determining the File Offset in the Case of Binary Files

In the case of binary files, the specification of the first date of the first channel is not made in bytes, but in *Records*, so that this value depends on the data type.

It should be noted that in the case of binary files too, the counting of the position numbers starts at **1**. The channel values are read from the data file starting from the starting point to be given in this field.

To skip comment lines in a binary file, the pointer to the first date of the first channel (file offset) is calculated as follows:

$$\text{Pointer} = \frac{\text{Commentblock in the channel}}{\text{No. of bytes of the datatype}} + 1$$

For example, to skip a binary header (comment block) of **512** bytes at the start of the channel to be read (data type of the channel: *INT**16***, i.e. the record size is **2** bytes), the following value should be specified for the channel offset:

$$\text{Pointer} = \frac{512\ \text{Bytes}}{2\ \text{Bytes}} + 1 = 257$$

❍ File Offset in the Case of ASCII Files

To skip leading comment lines in ASCII files, the pointer to the ASCII-line of the file which contains the first value of the first channel should be defined as follows:

File offset = no. of comment lines + 1

In the case of ASCII block files with CR/LF as the separator, the file offset should be specified for every channel n to be read:

File offset = no. of comment lines + nth channel

222: Channel offset

In the case of files which are saved in block format, the channel offset indicates the number of values between two successive values of the channel.

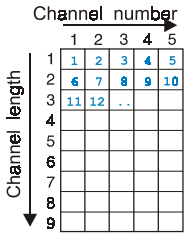
```
222,4
```

The following should be noted when specifying the channel offset:

- In the case of **data files stored channel-wise,** the channel offset is not required, since DIAdem can read the data with *data type*, the *number of values* and, if required, the *file offset* (if there is a header to be skipped).
- **In the case of binary block files, which do not contain any header to be skipped** and from which all the values are to be read, DIAdem can determine the channel offset from the specifications for *data type*, *number of values* and *file size* according to the following formula:

$$Offset = \frac{File\ size}{No.\ of\ values \cdot Type\ size}$$

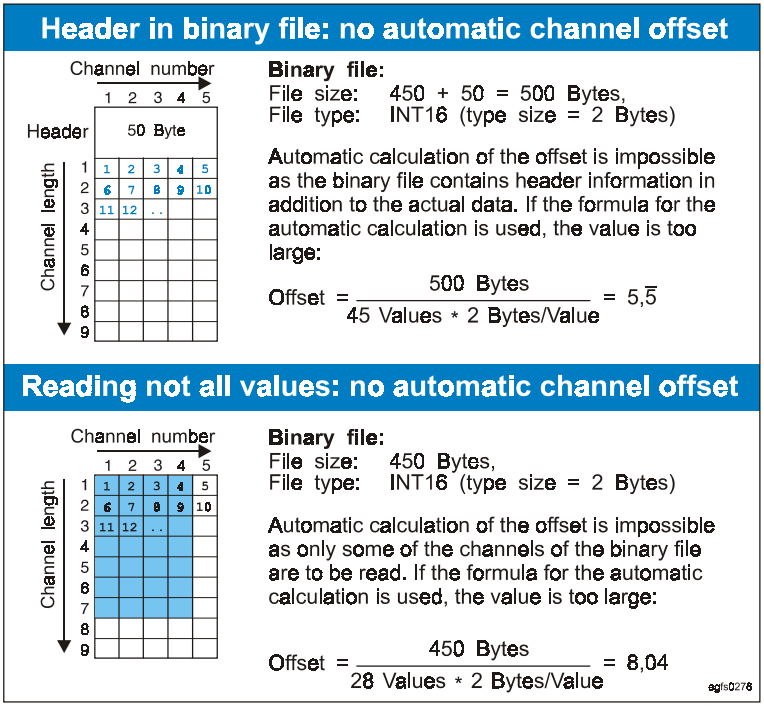
Automatic calculation of the channel-offsets



Binary file:
File size: 450 Bytes,
File type: INT16 (d.h. Type size = 2 Bytes)

$$Offset = \frac{450\ Bytes}{45\ Values * 2\ Bytes/Value} = 5$$

- **In the case of ASCII block files** with the separator *CR/LF*, the ***channel offset*** (the distance between two individual values of a channel) should be specified by the customer. DIAdem cannot then automatically determine how many channels the file-type contains, since in these files, the separator character and the character at the end of each line are the same (see. **diagram** in channel header entry 213).
- **In the case of binary block files which contain a header to be skipped** or all the values in which do not have to be read, DIAdem cannot determine the channel offset automatically. In such a case, the channel offset i.e. the number of records between two values of a channel, must be specified by the customer.

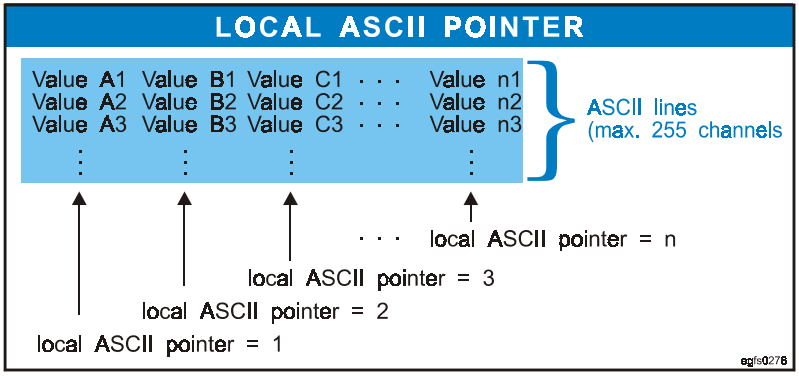


223: Local ASCII-pointer

To address or read a particular channel in an **ASCII block file,** it is necessary to know the position or rank of the value in the line which is assigned to the desired channel.

```
223,2
```

This information is stored in the local ASCII-pointer.



The local ASCII-pointer is **not** required for ASCII files stored channel-wise and for ASCII block files with *CRLF* as the separator.

230: Separator
In **ASCII-block files,** the data in the individual channels is delimited by **separator characters** (comma, spaces etc.). The separator character that exists in the ASCII-file can be different from file to file.

```
230,.
230,46
```

The separator character of the ASCII-file is defined with this entry. This is done by specifying the character itself or by specifying the decimal code of the character (e.g. 32 for a space). For ASCII-files which are saved block-wise, and have the **separator CR/LF** (decimal code 13/10), the character sequence CRLF should be entered as the separator in this entry.

```
230,CRLF
```

231: Decimals
ASCII-data files can have different numerical formats, depending on the type of storage.

```
231,,
```

This entry specifies the decimal character used in the channel, e.g. a comma. If no header entry is made here, DIAdem assumes that a point (dot) is being used for the ***decimal point*** **'.'**.

232: Exponential symbol
ASCII-data files can have different numerical formats, depending on the type of storage.

```
232,E
```

If no header entry is made here, DIAdem assumes that an **E** is being used for the ***exponential character.***

240: Starting value for the generation or offset
241: Step width for generation or scaling factor
The *starting value* and the *step-width* are used for the **definition of *implicit* data channels**. The *offset* and the *factor* are used for **scaling *explicit* data channels**. More details on *implicit* and *explicit* data channels can be taken from the entry *[210], channel type*.

```
240,0
241,1
```

❍ Starting Value and Step-Width in the Case of Implicit Channels

Implicit* data channels** are used when the channel contents can be generated from a **starting value** and repeated addition of a constant **step width**. DIAdem uses this facility for ***generating data channels with equidistant intervals.

The individual channel values can be calculated as

channel value(i) = starting value + (i-1)• step-width

with the counter **i** being incremented from *1* up to the *channel length*. The step width, in the case of the above *implicit* data, can be calculated from two consecutive values from the relation

step-width = channel value(i) - channel value(i-1)

The starting value is the 1st value of the channel to be read.

❍ Factor and Offset in the Case of Explicit Channels

In the case of ***explicit* data channels,** the factor and offset are used for **scaling** the saved data while loading in DIAdem. This method can be used, for example, for measurement data which initially often occurs as integer numbers or as voltage values within a certain voltage range. Scaling can also be applied to data which exists in real format.

A ***factor*** is used for conversion into the real physical readings, the raw data being multiplied by this factor. Next, an ***offset*** is added to the result of the multiplication.

When DIAdem reads the (measurement) data stored in the data file, it converts it using the offset and the factor by the following formula:

*channel value = offset + file value * calibration factor*

250: Minimum value of the channel
251: Maximum value of the channel
Minimum value/Maximum value of the data channel. If no value is entered here, the minimum value/maximum value is determined automatically by DIAdem when the data file is read.

```
250,0
251,59.84009999
```

252: Flag whether the channel contains NoValues
This flag indicates whether a channel contains ***NoValues***. *NoValues* are undefined values of a channel, e.g. missing readings or a missing series of measurements which could not be determined for some reason. The possible entries are *Yes* and *No.*

```
252,No
```

253: Flag whether the channel is monotone
This indicator provides *information* about the monotone behavior of the channel.

```
253,not monotone
```

The possible values are

- not monotone
- not calculated
- increasing
- decreasing

254: NoValue-value in the channel
This entry defines the *NoValue*-value which applies for this specific channel.

```
254,1.0E35
```

The values for **NoValues** can be defined either globally for the entire file (in the *general header*) or **individually for every data channel**. Only if the NoValue-is not specified in the channel header does DIAdem take the NoValue-value which has been set in the *general header*.

260: Form of display at the interface
Time-channels are stored like real channels. In order that such a channel be nevertheless displayed at the interface of the program, this entry must be set to Time.

```
260,Time
260,Numeric
```

270 … 275: Free real variables
Free real variables can be used for locating channel-specific quantities. They can be entered in the data area or read from the DIAdem variables `ChnAttrVal(i)`.

```
272,34.78654
```

280 … 284: Free Integer variables
Free Integer variables for storing the channel-related additional data. `ChnAttrInt(i)` is used for storing.

```
283,466
```

300 … 301: Reserve2
Free string variables for storing texts. The entry will be stored in `ChnAttrText(i).`

```
300,3D calculation
```

5 Examples of Data Headers

The following includes the general and channel-related data set descriptions for ASCII and binary files. DIAdem saves data in the binary format to save space. You can select other storage formats using the DIAdem DATA settings.

5.1 Examples of ASCII Files

This is a description of the structure of the data header for saving ASCII files in channels and blocks. When you import ASCII files, make these settings in the related Assistant. DIAdem then automatically generates the header file.

ASCII Block File

The following ASCII data file is to be read in DIAdem. This file is an ASCII file in block format with **6 channels** with 12 values each. Channel 3 is a **time channel** with the format *dd.mm.yyyy*. At the start of the file, there are **5 comment lines**, which are to be skipped. Blank lines are also considered to be comment lines.

```
1. line:    Example for reading an ASCII block file
2. line:    All channels which contain values which can be interpreted
3. line:    numerically are taken into the data area of DIAdem-DATA
4. line:
5. line:
6. line:    1st measurement   1    15.01.1996   6.00   2.1   3.34
7. line:    2nd measurement   2    15.01.1996   14.00  7.5   6.65
8. line:    3rd measurement   3    15.01.1996   22.00  5.7   4.98
9. line:    1st measurement   4    16.01.1996   6.00   1.3   2.37
10. line:   2nd measurement   5    16.01.1996   14.00  10.2  1.12
11. line:   3rd measurement   6    16.01.1996   22.00  5.9   2.69
12. line:   1st measurement   7    17.01.1996   6.00   3.4   3.72
13. line:   2nd measurement   8    17.01.1996   14.00  4.6   1.89
14. line:   3rd measurement   9    17.01.1996   22.00  0.5   6.47
15. line:   1st measurement   10   18.01.1996   6.00   2.9   9.15
16. line:   2nd measurement   11   18.01.1996   14.00  5     3.29
17. line:   3rd measurement   12   18.01.1996   22.00  4.4   1.54
```

Header File

The header file which is required to read the ASCII block file described above is reproduced below.

The comments in small print in the listing of the example file are only explanatory; they are not present in reality, and are, in fact, not allowed either!

```
DIAEXTENED  {@:ENGLISH
#BEGINGLOBALHEADER                                          (General header)
1,WINDOWS 32BIT                                             (Operating system)
2,{@R:800 {@V:8.00.981 {@F:4.00                             (Revision number)
101,Example for the header format DIAEXTENDED               (File comment)
106,
102,Example for reading an ASCII block file                 (comment)
106,
102,All channels which contain values which can be
106,
102, interpreted numerically are taken into the data area
106,
102, of DIAdem-DATA
106,
103,Zoe                                                     (Processed by)
104,01.08.2001                                              (Date of saving the file)
105,12:44:45                                                (Time of saving the file)
110,#dd.mm.yyyy                                             (Time format)
111,9.9000000000E+34                                        (value for NoValue in the file)
112,High -> Low
#ENDGLOBALHEADER

#BEGINCHANNELHEADER                                         (Channel header for channel 1)
200,Channel_No.1                                            (Channel name)
201,ASCII block file                                        (Channel comments)
202,-                                                       (Unit)
210,EXPLICIT                                                (Channel type)
211,asciiblk.asc                                            (Data file designation)
213,BLOCK                                                   (Method of storing the data)
214,ASCII                                                   (Data type)
220,12                                                      (No. of values in the channel)
221,1                                                       (Line with the 1st value in the channel)
223,1                                                       (Local ASCII-pointer to the column of the channel)
230,44                                                      (Separator character: space)
231,46                                                      (Decimal character: decimal point)
232,69                                                      (Exponential character: E)
240, 0                                                      (Offset)
241, 1                                                      (Scaling factor)
250,1                                                       Minimum value
```

```
251,3                                   Maximum value
252,No                                  NoValues
253, not monotone                       Monotone behavior
260,Numeric                             (Data display at the interface)
#ENDCHANNELHEADER                       (End of the channel header for channel 1)
```

All the entries in the first data channel are printed, and only items that vary are printed in the subsequent data channels.

```
#BEGINCHANNELHEADER                     (Channel header for channel 2)
200,Channel_No.2                        (Channel name)
                                        (Entries 201-221 identical to  those in the channel header 1)
223,2                                   (Local ASCII pointer to the column of the channel)
                                        (Entries 230-301 identical to those in the channel header 1)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                     (Channel header for channel 3)
200,Channel_No.3                        (Channel name)
                                        (Entries 201-221 identical to those in the channel header 1)
223,3                                   (Local ASCII pointer to the column of the channel)
                                        (Entries 230-254 identical with those in the channel header 1)
260,Time                                (Data display at the interface)
                                        (Entries 270-301 identical to those in the channel header 1)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                     (Channel header for channel 4)
200,Channel_No.4                        (Channel name)
                                        (Entries 201-221 identical to those in the channel header 1)
223,4                                   (Local ASCII pointer to the column of the channel)
                                        (Entries 230-301 identical to those in the channel header 1)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                     (Channel header for channel 5)
200,Channel_No.5                        (Channel name)
                                        (Entries 201-221 identical to those in the channel header 1)
223,5                                   (Local ASCII pointer to the column of the channel)
                                        (Entries 230-301 identical to those in the channel header 1)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                     (Channel header for channel 6)
200,Channel_No.6                        (Channel name)
                                        (Entries 201-221 identical to those  in the channel header 1)
223,6                                   (Local ASCII pointer to the column of the channel)
                                        (Entries 230-301 identical to those  in the channel header 1)
#ENDCHANNELHEADER
```

ASCII Block File

The following ASCII data file is to be read in DIAdem. This is an ASCII file in block format with **6 channels** with 12 values each. Channel 1 exists as a **time-channel** in the format *dd.mm.yyyy hh:nn:ss*.

```
15.01.1996 05:47:19, 1,  1,  6,  2.10, 3.34
15.01.1996 11:32:03, 2,  2, 14,  7.50, 6.65
15.01.1996 16:56:24, 3,  3, 22,  5.70, 4.98
16.01.1996 06:05:31, 1,  4,  6,  1.30, 2.37
16.01.1996 11:51:38, 2,  5, 14, 10.20, 1.12
16.01.1996 17:15:57, 3,  6, 22,  5.90, 2.69
17.01.1996 06:02:27, 1,  7,  6,  3.40, 3.72
17.01.1996 11:12:55, 2,  8, 14,  4.60, 1.89
17.01.1996 17:51:41, 3,  9, 22,  0.50, 6.47
18.01.1996 05:35:05, 1, 10,  6,  2.90, 9.15
18.01.1996 11:14:48, 2, 11, 14,  5.00, 3.29
18.01.1996 16:54:41, 3, 12, 22,  4.40, 1.54
```

Header File

The header file which is required to read the ASCII block file described above is reproduced below.

The comments in small print in the listing of the example file are only explanatory; they are not present in reality, and are, in fact, not allowed either!

```
DIAEXTENDED {@:ENGLISH

#BEGINGLOBALHEADER                                  (General header)
1,Windows 32Bit                                     (Operating system)
2,{@R:800 {@V:8.00.981 {@F:4.00                     (Revision number)
101,Reading an ASCII block file                     (File comment)
106,
102,ASCII block file with time channel              (Comments on data set)
106,
103,Zoe                                             (Processed by)
110,#dd.mm.yyyy hh:nn:ss                            (Time format)
111,9.9000000000E+34                                (Value for NoValue in the file)
112,High -> Low
#ENDGLOBALHEADER


#BEGINCHANNELHEADER                                 (Channel header for channel 1)
200,Time_Channel                                    (Channel name)
201,ASCII Block file                                (Channel comments)
202,-                                               (Unit)
210,EXPLICIT                                        (Channel type)
211,TIME_ASC.ASC                                    (Data file designation)
213,BLOCK                                           (Method of storing the data)
214,ASCII                                           (Data type)
220,12                                              (No. of values in the channel)
221,1                                               (Line with the 1st value in the channel)
```

```
223,1                          (Local ASCII pointer to the column of the channel)
230,44                                    (Separator character: comma)
231,46                                    (Decimal character: decimal point)
232,69                                    (Exponential character: E)
240,0                                                          (Offset)
241,1                                                  (Scaling factor)
250,6.2831051239E+10                                   (Minimum value)
251,6.2831350481E+10                                   (Maximum value)
252,No                                    (NoValues in the channel)
253,increasing                                              (Monotony)
260,Time                                  (Data display at the interface)
#ENDCHANNELHEADER              (End of the channel header for channel 1)
```

All the entries in the first data channel are printed, and only items that vary are printed in the subsequent data channels.

```
#BEGINCHANNELHEADER                (Channel header for channel 2)
200, Channel_No.2                                      (Channel name)
                  (Entries 201-221 identical to those in the channel header 1)
223,2                  (Local ASCII pointer to the column of the channel)
                  (Entries 230-242 identical to those in the channel header 1)
250,1                                                  (Minimum value)
251,3                                                  (Maximum value)
252,No                                    (NoValues in the channel)
253,not monotone                                            (Monotony)
260,Numeric                               (Data display at the interface)
#ENDCHANNELHEADER                  (End of the channel header for
                                                              channel 2)

#BEGINCHANNELHEADER                (Channel header for channel 3)
200, Channel_No.3                                      (Channel name)
                  (Entries 201-221 identical to those in the channel header 1)
223,3                  (Local ASCII pointer to the column of the channel)
                  (Entries 230-242 identical to those in the channel header 1)
250,1                                                  (Minimum value)
251,12                                                 (Maximum value)
252,No                                    (NoValues in the channel)
253,increasing                                              (Monotony)
260,Numeric                               (Data display at the interface)
#ENDCHANNELHEADER              (End of the channel header for channel 3)

#BEGINCHANNELHEADER                (Channel header for channel 4)
200, Channel_No.4                                      (Channel name)
                  (Entries 201-221 identical to those in the channel header 1)
223,4                  (Local ASCII pointer to the column of the channel)
                  (Entries 230-242 identical to those in the channel header 1)
250,6                                                  (Minimum value)
251,22                                                 (Maximum value)
252,No                                    (NoValues in the channel)
253,not monotone                                            (Monotony)
260,Numeric                               (Data display at the interface)
#ENDCHANNELHEADER              (End of the channel header for channel 4)
```

```
#BEGINCHANNELHEADER                          (Channel header for channel 5)
200,Channel_No.5                                          (Channel name)
                  (Entries 201-221 identical to those in the channel header 1)
223,5                 (Local ASCII pointer to the column of the channel)
                  (Entries 230-242 identical to those in the channel header 1)
250,0.5                                                   (Minimum value)
251,10.2                                                  (Maximum value)
252,No                                                (NoValues in the channel)
253,not monotone                                              (Monotony)
260,Numeric                                       (Data display at the interface)
#ENDCHANNELHEADER                 (End of the channel header for channel 5)

#BEGINCHANNELHEADER                          (Channel header for channel 6)
200, Channel_No.6                                         (Channel name)
                  (Entries 201-221 identical to those in the channel header 1)
223,6                 (Local ASCII pointer to the column of the channel)
                  (Entries 230-242 identical to those in the channel header 1)
250,1.12                                                  (Minimum value)
251,9.15                                                  (Maximum value)
252,no                                                (NoValues in the channel)
253,not monotone                                              (Monotony)
260,Numeric                                       (Data display at the interface)
#ENDCHANNELHEADER                 (End of the channel header for
                                                                  channel 6)
```

ASCII Channel File

The ASCII data file below is to be read into DIAdem. This is an ASCII file which has been stored in a channel-oriented manner. Totally, there are 6 channels present with 12 values each. Channel 3 is a **time-channel** with the format *dd.mm.yyyy hh:nn:ss*.

```
1
2
3
1
2
3
1
2
3
1
2
3
1
2
3
4
5
6
7
8
9
10
11
12
15.01.2001 00:00:00
15.01.2001 00:00:00
15.01.2001 00:00:00
16.01.2001 00:00:00
16.01.2001 00:00:00
16.01.2001 00:00:00
17.01.2001 00:00:00
```

```
17.01.2001 00:00:00
17.01.2001 00:00:00
18.01.2001 00:00:00
18.01.2001 00:00:00
18.01.2001 00:00:00
6
14
22
6
14
22
6
14
22
2.1
7.5
5.7
1.3
10.2
5.9
3.4
4.6
0.5
2.9
5
4.4
3.34
6.65
4.98
2.37
1.12
2.69
3.72
1.89
6.47
9.15
3.29
1.54
```

Header File

The header file which is required to read the ASCII channel file described above is reproduced below.

The comments in small print in the listing of the example file are only explanatory; they are not present in reality, and are, in fact, not allowed either!

```
DIAEXTENDED {@:ENGLISH

#BEGINGLOBALHEADER                                  (General header)
1,Windows 32Bit                                     (Operating system)
2,{@R:8.00 {@V:8.00.981 {@F:4.00                    (Revision number)
101,Reading an ASCII channel file                   (File comment)
106,
102,
103,Zoe                                             (Processed by)
104,01.08.2001                                      (Date of saving the file)
105,12:44:45                                        (Time of saving the file )
110,#dd.mm.yyyy hh:nn:ss                            (Time format)
111,9.9000000000E+34                                (Value for NoValue in the file)
#ENDGLOBALHEADER
```

```
#BEGINCHANNELHEADER                    (Channel header for channel 1)
200,Channel_No.1                                        (Channel name)
201,ASCII channel file                               (Channel comments)
202,-                                                           (Unit)
210,EXPLICIT                                          (Channel type)
211,asciikan.asc                                (Data file designation)
213,CHANNEL                              (Method of storing the data)
214,ASCII                                             (Data type)
220,12                                    (No. of values in the channel)
221,1                          (Line with the 1st value in the channel)
231,46                           (Decimal character: decimal point)
232,69                                (Exponential character: E)
240,0                                                      (Offset)
241,1                                                (Scaling factor)
252,No                                      (NoValues in the channel)
253,not monotone                            (Monotony of the channel)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER                  (End of the channel header for channel 1)
```

All the entries in the first data channel are printed, and only items that vary are printed in the subsequent data channels.

```
#BEGINCHANNELHEADER                    (Channel header for channel 2)
200,Channel_No.2                                        (Channel name)
                  (Entries 201-215 identical to those in the channel header 1)
221,13                         (Line with the 1st value in the channel)
                  (Entries 222-252 identical to those in the channel header 1)
253,increasing                              (Monotony of the channel)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                    (Channel header for channel 3)
200,Channel_No.3                                        (Channel name)
                  (Entries 201-215 identical to those in the channel header 1)
221,25                         (Line with the 1st value in the channel)
                (Entries 222-252 identical with those in the channel header 1)
53,increasing                               (Monotony of the channel)
260,Time                                    (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                    (Channel header for channel 4)
200,Channel_No.4                                        (Channel name)
                  (Entries 201-215 identical to those in the channel header 1)
221,37                         (Line with the 1st value in the channel)
                  (Entries 222-252 identical to those in the channel header 1)
253,not monotone                            (Monotony of the channel)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                    (Channel header for channel 5)
200,Channel_No.5                                        (Channel name)
                  (Entries 201-215 identical to those in the channel header 1)
221,49                         (Line with the 1st value in the channel)
                  (Entries 222-252 identical to those in the channel header 1)
```

```
253,not monotone                                  (Monotony of the channel)
260,Numeric                                       (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                               (Channel header for channel 6)
200,Channel_No.6                                  (Channel name)
                    (Entries 201-215 identical to those in the channel header 1)
221,61                                            (Row with the 1st. value in the channel)
                    (Entries 222-252 identical to those in the channel header 1)
253,not monotone                                  (Monotony of the channel)
260,Numeric                                       (Data display at the interface)
#ENDCHANNELHEADER
```

5.2 Examples of Binary Files

The structure of the header file and the relevant data file in the case of channel-wise and block-wise storage of binary files is explained below.

Binary Block File

The following binary data file is to be read in DIAdem. It is a binary file in block format with **5 channels** with 16,000 values each. The data is stored in 2-byte integer format (INT16).

Channel 1 serves as the time axis and is implicit, i.e. defined by specifying the starting value, the step width and the number of values.

The channels 2 to 5 exist explicitly.

```
1st value of channel 2
1st value of channel 3
1st value of channel 4
1st value of channel 5
2nd value of channel 2
2nd value of channel 3
2nd value of channel 4
...
```

Header File

The header file which is required to read the binary block file described above is reproduced below.

The comments in small print in the listing of the example file are only explanatory; they are not present in reality, and are, in fact, not allowed either!

```
DIAEXTENDED {@:ENGLISH

#BEGINGLOBALHEADER                          (General header)
1,Windows 32Bit                             (Operating system)
2,{@R:8.00 {@V:8.00.981 {@F:4.00            (Revision number)
101,Reading a binary block file             (File comment)
102,Format I16                              (20 Comments on data set)
103,Zö                                      (Processed by)
104,01.08.1996                              (Date of saving the file)
105,12:16:22                                (Time of saving the file)
110,#dd.mm.yyyy hh:nn:ss                    (Time format)
111,9.9000000000E+34                        (Value for NoValue in the file)
#ENDGLOBALHEADER


#BEGINCHANNELHEADER                         (Channel header for channel 1)
200,Time axis                               (Channel name)
201,t(s)                                    (Channel comments)
202,s                                       (Unit)
210,IMPLICIT                                (Channel type)
220,16000                                   (No. of values in the channel)
240,90                                      (Starting value)
241,0.001                                   (Scaling factor)
250,90                                      (Minimum value)
251,105.999                                 (Maximum value)
252,No                                      (NoValues in the channel)
253,increasing                              (Monotony)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER


#BEGINCHANNELHEADER                         (Channel header for channel 2)
200,P1                                      (Channel name)
201,Force                                   (Channel comments)
202,N                                       (Unit)
210,EXPLICIT                                (Channel type)
211,BINBLOCK.I16                            (Data file designation)
213,BLOCK                                   (Method of storing the data)
214,INT16                                   (Data type)
220,16000                                   (No. of values in the channel)
221,1                                       (Record with the 1st value in the
                                             channel)
222,4                                       (Channel offset)
240,0                                       (Offset)
241,0.0106811523                            (Scaling factor)
250,-124.07226512                           (Minimum value)
251,137.23144475                            (Maximum value)
252,No                                      (NoValues in the channel)
253,increasing                              (Monotony)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER
```

All the entries in the first data channel are printed, and only items that vary are printed in the subsequent data channels.

```
#BEGINCHANNELHEADER                          (Channel header for channel 3)
200,P2                                                        (Channel name)
201,distance vert.                                         (Channel comments)
202,mm                                                                  (Unit
                    (Entries 202-220 identical to those in the channel header 2)
221,2                                 (Record with the first value in the channel)
222,4                                                        (Channel offset)
240,0                                                                (Offset)
241,3.05176E-05                                                (Scaling factor)
250,-0.36181666560                                             (Minimum value)
251,0.36035182080                                              (Maximum value)
260,Numeric                                    (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                          (Channel header for channel 4)
200,P3                                                        (Channel name)
201,path horiz.                                            (Channel comments)
                    (Entries 202-220 identical to those in the channel header 2)
202,mm                                                                  (Unit
221,3                                 (Record with the first value in the channel)
222,4                                                        (Channel offset)
240,0                                                                (Offset)
241,1.525879E-04                                               (Scaling factor)
250, 1.1718750720                                              (Minimum value)
251,1.8457032384                                               (Maximum value)
260,Numeric                                    (Data display at the interface)
#ENDCHANNELHEADER

#BEGINCHANNELHEADER                          (Channel header for channel 5)
200,P4                                                        (Channel name)
201,Acceleration.                                          (Channel comments)
                  (Entries 202-220 identical to those in the channel header
                                                                           2)
202,m/sec2m                                                             (Unit
221,4                                 (Record with the first value in the channel)
222,4                                                        (Channel offset)
240,0                                                                (Offset)
241,3.051758E-04                                               (Scaling factor)
250,-7.4414067072                                              (Minimum value)
251, 1.4599610272                                              (Maximum value)
260,Numeric                                    (Data display at the interface)
#ENDCHANNELHEADER
```

Binary Channel File

The following binary data file is to be read into DIAdem. It is a binary file with the values stored in a channel-oriented manner. In all, **5 channels** with 16,000 values each in 2-byte integer format (INT 16) are stored in this file. Channel 1 serves as the time-axis and is implicit, that is, determined by specifying the starting value, the step-width and number of values. Channels 2 to 5 are present explicitly.

```
1 '1st value of channel 1 '
2
3
4
...
16000
1.003 '1st value of channel 2'
..
```

Header File

The header file which is required to read the binary channel file described above is reproduced below.

The comments in small print in the listing of the example file are only explanatory; they are not present in reality, and are, in fact, not allowed either!!

```
DIAEXTENDED {@:ENGLISH

#BEGINGLOBALHEADER                                  (General header)
1,Windows 32Bit                                     (Operating system)
2,{@R:8.00 {@V:8.00.981 {@F:4.00                    (Revision number)
101,Reading a binary channel file                   (File comment)
106,
102,Format I16                                      (Comments on data set)
106,
103,Zoe                                             (Processed by)
104,01.08.1996                                      (Date of saving the file)
105,12:15:25                                        (Time of saving the file)
110,#dd.mm.yyyy hh:nn:ss                            (Time format)
111,9.9E+34                                         (Value for NoValue in the file)
#ENDGLOBALHEADER

#BEGINCHANNELHEADER                                 (Channel header for channel 1)
200,Time axis                                       (Channel name)
201,t (s)                                           (Channel comments)
202,s                                               (Unit)
210,IMPLICIT                                        (Channel type)
220,16000                                           (No. of values in the channel)
240,90.000                                          (Starting value)
241,0.001                                           (Scaling factor)
250,90                                              (Minimum value)
251,105.999                                         (Maximum value)
252,No                                              (NoValues in the channel)
253,increasing                                      (Monotony)
260,Numeric                                         (Data display at the interface)
#ENDCHANNELHEADER                                   (End of the channel header for channel 1)

#BEGINCHANNELHEADER                                 (Channel header for channel 2)
200,P1                                              (Channel name)
201,force                                           (Channel comments)
202,N                                               (Unit)
210,EXPLICIT                                        (Channel type)
211,BINBLOCK.I16                                    (Data file designation)
213,CHANNEL                                         (Method of storing the data)
```

```
214,INT16                                   (Data type)
220,16000                                   (No. of values in the channel)
221,16001                                   (Record with the 1st value in the channel)
240,0                                       (Offset)
241,0.0106811523                            (Scaling factor)
250,-124.07226512                           (Minimum value)
251, 137.23144475                           (Maximum value)
252,no
253,increasing
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER                           (End of the channel header for channel 2)
```

All the entries in the first data channel are printed, and only items that vary are printed in the subsequent data channels.

```
#BEGINCHANNELHEADER                         (Channel header for channel 3)
200,P2                                      (Channel name)
201,distance vert.                          (Channel comments)
202,mm                                      (Unit)
                                            (Entries 203-220 identical to those in the channel header 2)
221,32001                                   (Record with the first value in the channel)
240,0                                       (Offset)
241,3.05176E-05                             (Scaling factor)
250,-0.36181666560                          (Minimum value)
251,0.36035182080                           (Maximum value)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER                           (End of the channel header for channel 3)

#BEGINCHANNELHEADER                         (Channel header for channel 4)
200,P3                                      (Channel name)
201,distance horiz.                         (Channel comments)
202,mm                                      (Unit)
                                            (Entries 203-220 identical to  those in the channel header 2)
221,48001                                   (Record with the first value in the channel)
240,0                                       (Offset)
241,1.525879E-04                            (Scaling factor)
250, 1.1718750720                           (Minimum value)
251, 1.8457032384                           (Maximum value)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER                           (End of the channel header for channel 4)

#BEGINCHANNELHEADER                         (Channel header for channel 5)
200,P4                                      (Channel name)
201,Acceleration.                           (Channel comments)
202,m/sec2                                  (Unit)
                                            (Entries 203-220 identical to those in the channel header 2)
221,64001                                   (Record with the first value in the channel)
240,0                                       (Offset)
241,3.051758E-04                            (Scaling factor)
250,-7.4414067072                           (Minimum value)
251, 1.4599610272                           (Maximum value)
260,Numeric                                 (Data display at the interface)
#ENDCHANNELHEADER                           (End of the channel header for channel 5)
```