
Subject: Re: Data Handling

Posted by [Harald Jeszenszky](#) on Fri, 24 Apr 1998 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello Neil,

I've tried to send you an E-mail several times but it has been returned by your server every time, so I answer you in the newsgroup:

```
> Harald
>
> Thanks for the help, it has been very useful. However, as I'm relatively
> new to IDL, some of the ideas have been hard to follow:
>
> IF (blocks EQ 1) THEN $ ; save block position
>   pos = (FSTAT(unit)).CUR_PTR $
> ELSE $
>   pos = [TEMPORARY(pos), (FSTAT(unit)).CUR_PTR]
>
> In the above piece of code I can't understand what "pos" does and where
> it is defined in the function.
>
> Many Thanks for your help
>
> Neil
```

I'm sorry for this rather complicated code but I'll try to explain it to you:

- The variable "pos" contains the positions of the data blocks within the data file.
- The function FSTAT(unit) returns an IDL structure containing status information of the file associated with the logical unit and the field CUR_POS reflects the current position of the file pointer. To get access to a field of a structure variable you have to use the '.' to reference the field.

The statement

```
pos = (FSTAT(unit)).CUR_POS
```

is a shorthand for:

```
info = FSTAT(unit)
```

```
pos = info.CUR_POS
```

- In IDL arrays can be concatenated by the "c=[a, b]" statement. The statement

```
pos = [pos, value]
```

adds one element to the variable "pos". If the initial value of "pos" is a scalar, the result is a 2-dimensional vector. The "TEMPORARY()" function is used to speed up the concatenation procedure and is useful for extending long arrays (see IDL Reference Guide or Online Help).

After reading the 1st data header (blocks = 1), the variable "pos" is initialized to a scalar containing the first data block position. Each new data block (blocks > 1) extends the variable "pos" by one element. If the data file contains "n" data blocks the variable "pos" results to an "n"-dimensional vector.

With kind regards
Harald
