

Starlink Project  
Starlink User Note 18.1

10.06.81

---

IDLPI - Convert Starlink image frames to IDL format

The purpose of IDLP1 is to convert 1 or 2-D Starlink frames into a format that can be read by the IDL system. Many different data formats can be converted to Starlink frame format by other utility programs, (e.g. IPCSIN, see SUN/3). Thus, IDLP1 forms a link which enables many different data formats to be converted into a format suitable for IDL. IDL is a powerful interactive data processing system that is installed on all Starlink computers. The main limitation of IDL is that it cannot handle variables larger than 16484 bytes. This limitation should be overcome when the VAX native mode version of IDL is released next year.

Program IDLP1 writes out data in the form of 2-byte signed integer arrays or 4-byte floating point arrays. The output format is selected by logical parameter 'INTEGER'. If INTEGER=Y an integer IDL array is written. If INTEGER=N a floating point IDL array is written. Program IDLP1 is a Starlink Application Program which can be run using the RUNSTAR command. The input frame is specified by parameter 'IN'. The output in IDL format is written to Fortran channel 10. A message describing the size of the array is sent to the terminal, together with a warning message if the array is greater than 16484 bytes in size. It is possible for IDL to process arrays greater than 16484 bytes in size by processing them one line at a time, so IDLP1 will continue writing arrays after the warning message is written in case you want to do this.

The use of IDLP1 is most easily illustrated by an example. Suppose you have some Starlink image frames in a directory called [IMAGES] and you want to convert some of them to IDL format for processing by IDL. The first thing you should do is to make [IMAGES] your default directory. Then set up the following command procedure in a file with name (say) IDLP1.COM

```
$ ASSIGN 'P1'.DAT FOR010
$ RUNSTAR SYS#SYSDISK:[USEREXE]IDLPI/IN='P1'
$ EXIT
```

You can now execute IDLP1 as follows:

## @IDLPI IMAGE

The parameter 'IMAGE' will cause IDLP1 to read a file IMAGE.BDF from your directory [IMAGES] and write a file IMAGE.DAT into it. Notice that the default file type 'BDF' is automatically supplied. All Starlink frames should have the file type 'BDF'. The program will prompt you for the format of the output image

### INTEGER/T/=

You can accept the default value of 'T' (which will cause an integer IDL array to be written), or you can reply 'F' (which will cause a floating point IDL array to be written).

Program IDLP1 will display the size of your image with the message:

```
AXIS1=mmmm
```

if the frame is 1-dimensional, or

```
AXIS1=mmmm  AXIS2=nnnn
```

if the frame is 2-dimensional. The symbols 'nnnn' and 'mmmm' will be replaced by the number of elements along that dimension. A 2-dimensional array will be written as 'nnnn' records containing 'mmmm' elements.

The array in 'IMAGE.DAT' can now be processed by IDL. Suppose that this is a 3x256 integer array (i.e. AXIS1= 3, AXIS2= 256). Then it can be read by IDL in the following manner:

First, set your terminal mode to upper case by typing

```
SET TERM/UPPER
```

Then, start up the IDL system by typing

```
IDL
```

The array can then be read by typing in the following commands in response to prompts ('IDL>' or '-') from IDL:

```
OPENR,1,'IMAGE'  
A=INTARR(3)  
IMAGE=INTARR(3,256)  
.RUN  
FOR J=0,255 DO BEGIN  
  FORRD,1,A  
  FOR I=0,2 DO IMAGE(I,J)=A(I)  
END  
END
```

This stores the values of the array in the IDL variable 'IMAGE'. This can now be manipulated and displayed using the facilities of IDL (see SUN/17). For example, the maximum value stored in the array can be displayed by typing in the commands

```
I=MAX(IMAGE) & PRINT,I
```

Note that if the array output by IDLP1 had been in floating point format you would have used the IDL function FLTARR instead of INTARR in the above example.

Program IDLP1 is a fully supported Starlink program and any problems with its use should be reported to Mike Lawden at Rutherford and Appleton Laboratory (Abingdon 21900 X6234, or username MDL or STAR). Program IDL is not a supported program. Any problems should be reported to Mike Lawden, however he cannot fix bugs in this proprietary package but only pass bug reports on to the firm who supplied it.

#### References

1. SUN/17 "The IDL language"