AIPS MEMO NO. 8

NATIONAL RADIO ASTRONOMY OBSERVATORY
Socorro, New Mexico

January 7, 1982

Memorandum

To: Post-Processing Group

From: R. Ekers

Subject: Map Word Types in AIPS

Although the use of I*2 word format for data storage in the post-processing system has been standard, I propose to change this to support either I*2 or R*4. In my view the main reasons for this change are to simplify application software and to improve its integrity. I consider the long term impact of this change very important since acceptance of the AIPS system by other users (especially users not involved in VLA reduction) will depend on the ease with which they can add or change application programs. Of course the other technical arguments must also be considered to make sure the effects of this change are acceptable.

Precision:

I*2 is sufficient for VLA maps and UV data but is insufficient in accuracy for beams and for some optical data. It is also insufficient for spectral line cubes where a single scale factor is used for the entire cube.

CPU Speed:

Most application programs will work in R*4, so two additional conversions are needed for the I*2 storage. This will only increase CPU time by a few percent so it is not an important criteria.

Disc Space:

I*2 storage halves the disk space required for data storage but a conversion program from R*4 maps to I*2 maps could be run separately to compact data when available space is critical.

I/O Speed:

This will be twice as fast for I*2 data. However, most programs which work in R*4 must write an output scratch file in R*4 to obtain the global minimum and maximum for rescaling to I*2. This actually gives 1.5 times more total I/O than for programs using storage in R*4 (I*2 in, R*4 out; R*4 in, I*2 out). An exception to this occurs for programs where the number of output maps (and hence scratch maps) are less than the number of input maps (e.g., a program to form the mean from a set of line channels or a program to extract a few parameters from an entire map).

Software Costs

It has long been reconized that in almost any significant computer system the software costs exceed hardware costs. Consequently we must consider the software implications as well as the hardware implications. The simplest question to evaluate is the direct effect on software implimentation. Perusal of existing software indicates that 10% of the code could be eliminated by use of R*4 throughout. The present post processing software effort has cost about $500K so the saving from software effort alone would have been about $50K.

Integrity, Debugging and Maintenance:

All three are significantly worse for I*2 software because of the increased program complexity and the need to check for overflow and truncation. It is hard to make a quantitative evaluation of the long term effect but I would judge these to eventually become the most important factors.

Summary:

The two arguments against R*4 storage are the cost of total disk space and the extra time which some programs will require for I/O. Table I gives the cost breakdown for the VLA VAX post-processing system. In the worst case, we must double the disk space for R*4 storage at an incremental cost of $40K or approximately one-man year of software effort. In terms of the relative cost, this is an increase of 10% which seems a small price to pay for the simplification of all future software development. It could be argued that the software cost occurs only once, but this assumes a limited and static set of software. Also the $40K increase is the worst case, if we assume that half the data is converted to I*2 and that a typical outside users system will have only 1/2 our storage capacity, we will only increase the cost by about $10-15K.

If the I/O capactiy is the limiting factor, we can add an additional unibus adapter at a cost of $12.3K.

Suggested Method of Implementation:

The map input interface should make the word type transparent to the application program. It should deliver data to the program in R*4 (optionally I*2) - Walter Jaffe's I/O routines do this nicely already. The standard I/O output should be R*4, but as the last step in the program a switchable module should be called to convert the map back to I*2, and delete the R*4 version.

It is important that we begin this change immediately since the longer we wait the harder it will be to implement. I propose to use the following scheme to minimize impact on current software development.

1. All new application software should use Jaffe's I/O.

2. The default output switch should be initially set for I*2.

3. Slowly change existing application software to use Jaffee's I/O (or at least to accept R*4). These changes could be done as changes are needed for other reasons.

4. When this conversion process is sufficiently advanced, change the default output switch to R*4 and supply a user procedure to convert to I*2. Then leave it to the user to use this procedure to convert data for any remaining programs which are still reading I*2.

Other Data Structures:

In this memo I have only considered the relative merit of R*4 and I*2. However, there is an alternative in which a scale factor is included in every line of I*2 data. This makes it trivial for standard I/O conversion to I*2 and can overcome all objections to I*2 storage except for CPU speed and accuracy. Possibly it should also be considered and if so the implementation should include in the 2D header an indication of the length of 1D header (likewise between 3D and 2D). Considerable generality in the data base can be achieved in this way. The obvious objection to this scheme is that the change in database structure required has much more impact on existing software.

RDE/er
Enclosure

TABLE 1: Cost Breakdown for Post Processing VAX

	Present System $K	Double Mass Storage and I/O Rate $K
VAX + 2 M byte + TU77 + RP06	205	---
2M byte Memory	10	---
Tape dirve	30	---
3 Century disks at $12.3K	36.9	+ 36.9
Disk controller (Max 4 disks each)	4.2	+ 4.2
Unibus Adapter	12.3	+ 12.3
I2S Display	45	---
Array Processor	75	---
Terminals, Plotter	30	---
TOTAL	448.4	53.4