SGP MEMO NO. 43

MEMO

TO: Pipeline Software Group
FROM: Jim Torson
SUBJECT: FORTRAN Standard Command Package
DATE: December 9, 1983

In response to a request by Don Retallack, the following is a list of work that has occurred to me which might be done on the FORTRAN Standard Command Package:

1. There is a minor bug in the NEEDCH routine. If you supply a string that is longer than the requested string, it will produce a string of the requested length in which the last character is obtained from the end of the supplied string. In other words, it doesn't just truncate the input string to the requested length. Also, if you supply a string that contains a quotation mark followed by some blanks followed by the string of interest, then you get a strange result.

2. Command names should be standardized. For example, CATLST uses the CAT-ID command to specify the user number, but TVLOD uses the CATALOGUE-ID command. For the new specification of the disk pack on which to find a map, we have implemented PACK commands. Would a better name be PACKNAME, DISK, etc.?

3. When the user types INPUTS, the CATALOGUE-ID command gives something of the form nnnn.CAT. Perhaps the ".CAT" should be dropped. (The CAT-ID command in CATLST doesn't give the ".CAT".) Also, perhaps leading zeros should be dropped since the user doesn't need to type them when he enters the command.

4. There are a number of new commands that have been implemented that should be made part of the built-in commands. These include PACK, NAME, CLASS, and VERSION. Currently, each program uses its own copy of the code for each of these commands.

5. Currently the SAVECOMMANDS file is unique according to the terminal number and the first number of the UIC. Since everybody runs from the same UIC ([300,20]), this doesn't provide the desired separation between different users. Perhaps we should change this scheme.

6. When you run a program, it takes longer than would be desirable for the program to initialize itself and begin accepting commands. I suspect that this is due to reading and interpreting the SAVECOMMANDS file. Perhaps this could be speeded up in some way. For example, a binary file that doesn't need to be scanned and interpreted could be written. The text version of the file could still be retained for use when the user explicitly types SAVECOMMANDS or GETCOMMANDS.

7. It might be useful to add a standard command which would exit from the current task and start up another specified task.

8. Perhaps it would be useful to consider putting the code for the the standard commands into a separate task. This could reduce the need for overlays in the applications programs and speed their task building.

There are some potential problems though. For example, what do you do with commands that are unique to just a single task?

9. A task will typically use some of the built-in standard commands plus some commands that are unique to the task. Perhaps we should establish some conventions for naming of the routines which implement the task-specific commands. For example, should the routine name start with SC like the built-in commands or just be some six-character abreviation of the command name? Also, should each of these be stored in a separate souce file or should they all be contained in a single file which has a name that is related in some way to the name of the task in which they are used? (Currently some programs use one convention and some the other convention.) Also, a task will typically use a special version of the SCPARM routine which is tailored according to the built-in commands that the task actually uses. Perhaps there should be a convention for the name of the source file which contains this routine. Ditto for the SCXTRA routine. (If the code for all the unique commands in a task are stored in a single source file, perhaps SCXTRA should be stored there also.)