

## **OTHER Users**

### **Overview**

TART2002 is so computer independent that you will find it is almost trivial to install and use it on any type of computer. Here I will describe in detail the few remaining computer dependent parts of the code. Once you have supplied these for your type of computer you can proceed to install using either the UNIX (32 bit computer) or CRAY (64 bit computer) installation instructions.

### **Computer Dependence**

The only capability that you **MUST** have to use TART2002 is CRAY like POINTERS, e.g.,

```
POINTER(LOCX,X)
```

to assign dynamic memory. If you don't have this capability, sorry about that, but you cannot use TART2002.

TART2002 is written in very strict FORTRAN and you should be able to use it on virtually any computer. The few remaining parts of the code are isolated in the module ???f, where ??? defines each type of computer, e.g., DEC.f, SUN.f, HP.f. These features include, assignment of dynamic memory, defining time and date, and reading parameters from the input line; nothing else. Of these the only one that is **REQUIRED** is assignment of dynamic memory. As will be explained below, all of the others can be satisfied by dummy routines.

In order to implement the code on other computers you should be aware of the following computer dependence in ???f; you can then define the equivalent for your computer and implement the system. I recommend that you start with SUN.f and modify it to meet your needs. There are only four computer dependent routines in each of the ???f files.

### **For Starters**

Try using SUN.f without any changes. If everything loads o.k. you are finished. However, if your loader tells you that any routines are missing you can use the following guidelines to define the missing routines.

As you will see below, the only necessary routine is the memory manager. All the others can be replaced by dummy routines (described in detail below). However, it is **HIGHLY RECOMMENDED** that you include a timing routine, so that you know how long the code takes to execute. **EVERY** computer has a means of defining execute, or real, running time; see the various ???f files for how this is done on a variety of computers. For more details see more extensive remarks on timing at the end of this section.

**Subroutine MEMLOW**

Memory manager = 32 vs. 64 bit pointers: so far the only workstation that we have found that uses 64 bit pointers is the DEC-Alpha. So in most cases you can use the routine from SUN.f. If your computer does use 64 bit pointers use the routine from DEC.f. If you are not sure how your computer treats pointers, try the 32 bit version from SUN.f.

For dynamic memory allocation this routine uses **MALLOC**, to assign memory, and **FREE**, to free memory. If the names of these routines are different on your computer, you need merely change **MALLOC** and **FREE** to the names used on your computer.

If your system does not have the equivalent of **MALLOC** and **FREE**, use the MALLOC.f routine supplied with the UNIX version.

Note, of the routines discussed here this is the only one that **MUST** be defined for TART2002 to run properly.

**Subroutine INLINE**

Reads the input command line to optionally define the names of the TART input and output files. If your computer does not have a way to do this - no problem - just add a dummy routine to immediately return. This will limit you to **ALWAYS** using the default input and output filenames TART.IN and TART.OUT, which is what I recommend anyway.

```
SUBROUTINE INLINE(TARTIN,TARTOUT)
RETURN
END
```

**Subroutine TIMEIT**

Define the execution time since the problem started running. Every computer has a way to do this, but it varies widely from computer to computer; see the other ???f files for examples. It is best if you can define EXECUTION (CPU) time, but if you can't use REAL (Clock) time - one or both of these can be defined on EVERY computer. If all else fails and you cannot define any form of time - no problem - TART2002 will still execute correctly if you just add a dummy routine to always define time as 0,

```
SUBROUTINE TIMEIT(SECONDS)
include 'implicit.h'
SECONDS=0.0
RETURN
ENTRY TBASE
RETURN
END
```

Note, this is not recommended, since you will not be able to time the execution of the

codes; again, see more extensive remarks on timing at the end of this section.

**Subroutine DATEIT**

Define the time, data and computer in characters - 8 characters for each, for example,

```
12:00:00  
03/30/03  
SUN
```

Most computers have a way to do this, and it is useful documentation since it appears in your TART2002 output file, which allows you to later determine exactly when any given problem was run; see the ???f files for examples how this is done on a variety of computers. If your computer doesn't have a way to do this - no problem - just add a dummy routine to return any reasonable values that you want to appear in your output listing,

```
SUBROUTINE DATEIT(ARRAY)  
CHARACTER*8 ARRAY(3)  
ARRAY(1) = '12:00:00'  
ARRAY(2) = '03/22/03'  
ARRAY(3) = 'SUN   ' ! here define your computer type  
RETURN  
END
```

**That's it!!!!**

These are the only remaining computer dependent routines in TART2002. Once you have defined an equivalent ???f for these four routines for your computer you will be able to use the code.

If you do develop a ???f for a different type of computer please return a copy of me - it will be included in future distributions of this code - which will make your work easier at that time.

You should next read the UNIX (32 bit) or CRAY (64 bit) portion of this installation manual and follow the steps to implement this system; at each step you can look inside the MAKEIT.BAT, etc. files to see what they are doing. You will see that they are merely compiling codes and executing them, and from the Makefiles for TART2002, etc. you will be able to see which files to compile and load together. Once you understand what these files are doing you should be able to do the same thing on your computer.

## **Timing Execution**

Most UNIX systems allow ETIME to be used to time execution, but there are a number of other routines - here's a list,

ETIME(T) - SUN, SGI, MEIKO, DECALPHA, IBM-PC, POWERMAC  
SECOND(0) - CRAY  
CLOCK() - IBMRISC  
TIME(0) - HP  
SECNDS(0.0) - VMP

Individual routines are defined for all of these computers. If you have a different type of computer that doesn't use one of these routines I recommend that you use the SUN timing routine using ETIME and add your own defined ETIME routine at the end of the FORTRAN deck,

```
FUNCTION ETIME(TARRAY)
  DIMENSION TARRAY(2)
  TIMENOW = ???????
  TARRAY(1) = TIMENOW
  TARRAY(2) = 0.0
  ETIME = TIMENOW
  RETURN
END
```

Where ??????? is the name of the routine on your system that defines time (execution or real time) in seconds - either from the start of execution or some fixed time - it doesn't matter, since TART2002 will only use differences in time defined by successive calls to ETIME.