Data Analysis Applications Group (DAAG) Update: Retrieve MDSplus data in Fortran 90/95

Justin Burruss
2004.04.22

You can retrieve MDSplus data in Fortran 90/95

- It is possible to retrieve MDSplus data using Fortran 90/95
- Very useful if you need to crunch numbers using MDSplus data
- Last week we laid the groundwork by demonstrating dynamic memory allocation
- Today we look at how to get data from MDSplus

Connect to MDSplus server using MdsConnect

- Use MdsConnect to connect to Atlas

```
status=MdsConnect('atlas')
if (status .eq. -1) then
   stop 'Could not connect to server'
endif
```

- A return value of -1 indicates error
- This is just like using MdsConnect in IDL or C.

After connecting, open the tree using MdsOpen

- In this example, we're opening the EFIT01 tree for shot 111203

```
status=MdsOpen('efit01', 111203)
if (mod(status,2) .eq.  0) then
   stop 'Could not open tree'
endif
```

- An even number return value indicates an error
- This is just like using MdsOpen in IDL or C.

Use allocate because we don’t know data size until runtime

- Use MdsValue to get the length of the data, in this case AMINOR
- Then use allocate to allocate space for AMINOR

```
c Find size of AMINOR
      dsc = descr(IDTYPE_LONG,len,0)
      status=MdsValue('SIZE(\AMINOR)', dsc, 0, 1)
      if (mod(status,2) .eq. 0) then
         stop 'Could not get length of AMINOR'
      endif

c Allocate space for AMINOR (and the timebase)
      allocate(aminor(1:len), t_aminor(1:len), stat=err)
      if (err .ne. 0) then
         stop 'Could not allocate memory for AMINOR'
      endif
      dsc = descr(IDTYPE_FLOAT, aminor, len, 0)
      t_dsc = descr(IDTYPE_FLOAT, t_aminor, len, 0)
```

Use MdsValue to get the data

- Use MdsValue to get the data, in this case AMINOR
- In this example we also get the timebase of AMINOR

```
c Retrieve AMINOR
      status=MdsValue('\AMINOR', dsc, 0, 1)
      if (mod(status,2) .eq.  0) then
         stop 'Could not get AMINOR'
      endif

c Retrieve timebase of AMINOR
      status=MdsValue('DIM_OF(\AMINOR,0)', t_dsc, 0, 1)
      if (mod(status,2) .eq.  0) then
         stop 'Could not get AMINOR timebase'
      endif
```

Deallocate memory when it is no longer needed

- After you are done crunching numbers, free unneeded memory using deallocate

- Efficient use of memory means that our computers run faster

```
c      we're done, so deallocate memory
       deallocate(aminor, t_aminor, stat=err)
       if (err .ne. 0) then
          stop 'Could not deallocate memory for AMINOR'
       endif
```