

This file extends the FM_User_Manual.txt file for the modules fm_quadreal.f95, fm_doubleint.f95, and fm_quadint.f95. These modules allow programs to work with FM when they explicitly declare some variables to be quad real, quad complex, double integer, or quad integer.

The operations defined by these modules include assignments ($a = x$), comparisons ($a < x$), and arithmetic ($x = x + a$), where one of a, x is type(fm), (im), or (zm), and the other is quad real, quad complex, double integer, or quad integer.

All of the programs here use the 3 files discussed in the basic FM_User_Manual:

1. fmsave.f95 Module for FM internal global variables
2. fm.f95 Subroutine library for multiple-precision operations
3. fmzm90.f95 Modules for interfaces and definitions of derived-types

These are the three modules:

4. fm_quadreal.f95 Module for interfacing FM with quad precision real or complex variables
5. fm_doubleint.f95 Module for interfacing FM with double length integer variables
6. fm_quadint.f95 Module for interfacing FM with quad precision integer variables

There is a test program to check the operations defined in each of these modules:

7. TestFMquadReal.f95 Test program checks operations for each of the fm_quadreal routines
8. TestFMdoubleInt.f95 Test program checks operations for each of the fm_doubleint routines
9. TestFMquadInt.f95 Test program checks operations for each of the fm_quadint routines

There is a small sample program illustrating the use of the operations defined in these 3 modules. The chk file shows some typical output from the program. Other compilers or computers may differ.

10. SampleDQ.f95
11. SampleDQ.chk

After the first 3 files have been compiled and the program TestFM.f95 has been compiled and run to verify the basic FM package has been installed successfully, as shown in FM_User_Manual.txt, files 4 through 10 can be compiled and run.

The interface operations are made available in the user's program by putting one or more of these statements

```
use fm_quad_real
use fm_double_int
use fm_quad_int
```

at the top of routines using one or more of these variables if they interact with (fm), (im), or (zm) variables.

Here are sample commands to run the programs in files 7 through 10. These are for the gfortran compiler on a Windows PC. Macs are very similar, as are other compilers. See FM_User_Manual.txt.

```
gfortran fm_quadreal.f95 -c -O3
```

```
gfortran TestFMquadReal.f95 -c -O3
```

```
gfortran fmsave.o fm.o fmzm90.o fm_quadreal.o TestFMquadReal.o -o TestFMquadReal.exe  
./TestFMquadReal
```

```
gfortran fm_doubleint.f95 -c -O3
```

```
gfortran TestFMdoubleInt.f95 -c -O3
```

```
gfortran fmsave.o fm.o fmzm90.o fm_doubleint.o TestFMdoubleInt.o -o TestFMdoubleInt.exe  
./TestFMdoubleInt
```

```
gfortran fm_quadint.f95 -c -O3
```

```
gfortran TestFMquadInt.f95 -c -O3
```

```
gfortran fmsave.o fm.o fmzm90.o fm_quadint.o TestFMquadInt.o -o TestFMquadInt.exe  
./TestFMquadInt
```

```
gfortran SampleDQ.f95 -c -O3
```

```
gfortran fmsave.o fm.o fmzm90.o fm_quadreal.o fm_doubleint.o fm_quadint.o SampleDQ.o -o SampleDQ.exe  
./SampleDQ
```

The three TestFM... programs test all the operations defined in each module. Run the ones your program will use to make sure that the modules have been successfully installed. At the end of each run, it should say:

```
... No errors were found.
```

SampleDQ is a small program showing how to do some calculations combining (fm), (im), and (zm) variables with variables of type quad real, quad complex, double integer, or quad integer.