

Overview of TMG User Subroutines

I-DEAS TMG is a sophisticated heat transfer simulation code that is rich in features. However, there are still times when additional features are needed. There are also occasions when a user requires TMG to share data with another simulation code. Fortunately, TMG includes features that make these tasks relatively easy.

The *TMG Open Architecture Primer* is a guide to TMG open architecture implementation. MAYA has used these features many times to enhance TMG and this document outlines some preferred methods of implementation. It is not essential that these methods are followed in order to successfully use TMG's open architecture features. However, they are based upon considerable experience, and so their use is strongly recommended. The reference document for TMG open architecture is the *TMG Reference Manual*. User written subroutines are addressed in the section *Card 10 - User-Written Subroutines USER1 and USERF*.

The balance of this article is a general discussion of TMG user subroutines intended to help you decide whether or not to use this feature. Refer to the *TMG Open Architecture Primer* for more detailed instructions. The *TMG Reference Manual* should also be consulted for detailed specifications. Both these documents are available on the I-DEAS Online Help Bookshelf (click on the Bookshelf icon at the top of this page).

USER1 is an optional FORTRAN 77 user-written subroutine that is automatically compiled and loaded with the Analyzer at run time. USER1 may call other subroutines. All user-written subroutines and user-defined common blocks should be called USER2, USER3, ..., to prevent name conflicts with the subroutines of the Analyzer.

There are many applications for *User Subroutines*, including:

- integration of external code (e.g. to compute heat transfer coefficients)
- alternative correlations for fluid flow or convection
- application of complex boundary conditions
- customized output (including I-DEAS data sets)
- active temperature control systems
- time-varying spatial domains (ablation, phase change)
- sequential TMG runs

You must have a Fortran compiler to use this option. The following table contains a listing of recommended compilers for different platforms that are valid for compiling User Subroutines for TMG 12.0.0. Later versions of the compilers are also supported.

PLATFORM	COMPILER
IRIX	MIPSpro Version 7.4.2m
HP-UX	HP F90 version 2.6 or later
Solaris	Sun Workshop 6 Fortran 77 version 5.2 or later

Linux	Portland Group Fortran 77 version 5.2-4 or later
IBM-AIX	FORTRAN 8.1 (but C / C++ from Visual Age 6)
Windows	Compaq / HP Visual FORTRAN 6.0 (now called Intel Fortran Compiler)

If you are using a patched version of TMG, you can check for any changes or updates to the recommended compilers by looking at the following file. Use your web browser to go to the root directory of your I-DEAS installation, then open this file:

.../tmg/if/tmg_new.html

Including a User Subroutine

To include a User Subroutine in a TMG model run, you must

1. Create a file which contains USER1 and all of the routines which it calls. The USER1 routine must come first in the file.
2. Define the file as an *Include File* on the *TMG Study Setup* form. Alternately, you can define a file that names up to 30 other files to include. Included files can be either user subroutines or files in the TMG INPF file format.

To create a file that names other files to include, use the following format:

```
INCLUDE FILE LIST
filename1
filename2
...
```

where filename* represents the name of a file in the current run directory or working directory, or the path to any file (total path and file name length cannot exceed 80 characters).

When the model is solved, the contents of the include file will automatically be detected as FORTRAN code, and included in the TMG input file in the User Subroutine card block. The routines will then be automatically compiled and linked with TMG's solver module. The commands which carry this out are contained in a script file `.../tmg/exe/adduser1.com.`; you can copy this file to your local directory and customize it if desired.

Using Group Names

To develop a User Subroutine which is general purpose (i.e. applicable to more than a single model), you can use element groups. The NAMAR routine described below allows you to resolve the contents of an I-DEAS group from within USER1. In this way, you can develop the subroutine so that it is not dependent on element labeling, and can therefore be used for any model which includes the appropriate element groups.

Note that the TMG input file convention limits group names to 7 characters, with no embedded blanks. The name must start with a letter, and may not include special characters. In generating a TMG input file from an I-DEAS model, any group names which do not conform to this convention are changed to do so. Therefore, if you are developing a general purpose User Subroutine based on element groups, be sure to use

group names which conform to the INPF convention.

Using Element Labels

Each element in the model is assigned an internal element number in the TMG solver. Most of the model parameters are accessed by referring to this internal element number. Use the function **KCONV** (see below) to determine internal element numbers.

Calling Other Subroutines

You can call other user-supplied subroutines from **USER1**. To prevent naming conflicts with routines within TMG's solver, it is safest to give any such routines the names **USER2**, **USER3**, etc. They should be added to the *Include File* after **USER1**.

File Access

You can access all files from **USER1**. Their pre-assigned numbers are:

TEMPF	Unit 2
INPF	Unit 89
MODLF	Unit 7
VUFF	Unit 8
FMODLF	Unit 16
MSGF	Unit 17
REPF	Unit 18
MODLCF	Unit 56
GTEMPF	Unit 62

INPF should not be modified.

New files created with an **OPEN** statement should be numbered 20-40.

USER1 Format

The header of the **USER1** subroutine that you create must be as follows:

```

SUBROUTINE
+USER1 ( GG , T , C , Q , QD , R , TIME , DT , IT , KODE , NOCON , MAXNO ,
+ICONV , DUM1 , DUM2 , DTP , TF )
  DIMENSION GG ( * ) , QD ( * ) , T ( * ) , C ( * ) , Q ( * ) , R ( * ) , ICONV ( * )
  CHARACTER*7 NAME
  COMMON/TDMAX/TDMAX
  COMMON/PRTFLG/PRTFLG
  COMMON/IRUN/IRUN , IR ( 1 )
  COMMON/MAXNOQ/MAXN1 , MAXN2
  COMMON/PARAMS/PARAMS ( 80000 )
  COMMON/GRAV/GRAV , GV ( 3 ) , TABS , RGAS , PSTD , TSTD , SIGMA
  SAVE
  |
  |
  RETURN
  END

```

Calling Sequence

1. Analyzer Initialization - all conductances, capacitances, etc. are read in and placed into the proper arrays.
2. If Card 9 PARAM USRCNDADD is present, USER1 is entered with KODE=5. If CALL ADDCOND is present, new conductances are added to the matrix.
3. If hydraulic elements are present, the hydraulic iteration loop is entered. in this loop USER1 is entered once per hydraulic iteration with KODE=4, during which you can modify total pressures and hydraulic flow resistances.
4. Analyzer evaluates conductances, performs table interpolation, and evaluates heat loads.
5. USER1 is entered with KODE=1. You can modify conductances, capacitances, heat loads, sink temperatures, and other parameters which will be used for temperature calculation.
6. The Analyzer calculates temperatures for the next iteration or time step.
7. USER1 is entered with KODE=2. You can modify the calculated temperatures.
8. If the run is complete go to step 9.

If the run is not complete go back to step 3.
9. USER1 is entered with KODE=3 and IRUN is set to IRUN+1. Check if IR (IRUN) is set or not.

If IR(IRUN) is not set go to step 10.

If IR(IRUN) is set go back to step 3.
10. Run is completed.

Copyright (c) 2007 UGS Corp. All Rights Reserved.