Run CMD Window Guide

for the

MYSTRAN General Purpose Finite Element Structural Analysis Computer Program

(Open Source Version)

Dr. Bill Case – Original Author

Subsequent Authors:

Brian Esp

Zach Lerner

Victor Kemp

Steve Doyle

Bruno Paschoalinoto

MYSTRAN

www.mystran.com

github.com/MYSTRANsolver

(September 2025)

Most consistent with MYSTRAN program version 17.0

1 RUN CMD WINDOW GUIDE

This guide requires a complete rework. It is a place holder for now.

2 Command window messages

When you ran the example problem in the command window (or terminal) you should have seen messages such as:

>> MYSTRAN BEGIN:

>> LINK i BEGIN

.

.

.

>> LINK i END

>> MYSTRAN END:

with several lines in between indicating the progress as MYSTRAN solves your problem. The MYSTRAN program is divided into several major subroutines (called LINK's) that solve the problem. Below is a list of these LINK's and what they do:

- LINK0:

o Read input data deck and check for errors and possible restart

o Process grid and coordinate system input data

o Process Case Control output requests

o Forms degree of freedom (DOF) tables

o Process concentrated mass input data

o Calculates rigid body mass properties (Grid Point Weight Generator)

o Process temperature and pressure load input data intp arrays needed for element load calculations

- LINK1:

o Process MPC's and rigid elements into sparse G-set array RMG

o Process all applied forces (including grid forces and moments, gravity, pressure, thermal, centrifugal, scalar) into sparse G-set load array PG

o Formulate the G-set sparse stiffness and mass arrays KGG and MGG

o Formulate the G-set sparse differential stiffness array KGGD

 LINK2:

o Reduce the G-set stiffness, mass, load and constraint matrices to the L-set

 LINK3 (for statics problems only):

o Solve for the L-set displacements

 LINK4 (for eigenvalue problems only):

o Solve for the L-set eigenvalues and eigenvectors

 LINK5:

o Build the A-set displacements back up to the G-set through use of the constraint matrices

 LINK6 (for Craig-Bampton model generation only):

o Builds a Craig-Bampton model from the input physical model

 LINK9:

o Use the G-set displacements from LINK5 to solve for the outputs requested in Case Control.

If the job executes without error the last message from MYSTRAN (after >>>>> LINK 9 END) is:

>> MYSTRAN END : 1/19/2006 at 15: 5: 3.8. The output file is:

filename.F06

MYSTRAN terminated normally. Total CPU time = 1.56E-01 seconds

where filename is the name of the input file

3 MYSTRAN files

MYSTRAN produces several output files besides the F06 file (which contains the output of the answers to the problem). If filename.dat is the input file name, the output files are all filename.ext

where ext is the extension of the output file. Each of the output files is discussed below under the heading of the extension for that file.

3.1 F06 text output file

The F06 output file has the answers for the problem submitted to MYSTRAN. It is a text file that can contain an echo of the input, messages that MYSTRAN outputs (warning, fatal error and information messages) as well as the outputs that were requested in Case Control. Appendix A of the MYSTRAN Users Reference Manual shows the F06 output file for example problem: 1 EXAMPLE1.F06).

3.2 F04 text output file

The F04, or log, output file can contain a list of the subroutines run along with the subroutine begin and end times. This text file is only produced if explicitly requested. That request must be made using the MYSTRAN initialization file discussed in a later section.

3.3 ERR text output file

A list of all warning and fatal error messages is printed in the F06 file as discussed above. However, since warning messages may be suppressed in the F06 file via the Bulk Data PARAM SUPMSG entry, a separate list of all warning and error messages is put into the ERR text file as well. This file only contains the warning and fatal error messages.

3.4 BUG text output file

If element debug information is requested in Case Control via the ELDATA command, a text file with extension BUG is created. The contents of that file are enumerated in the MYSTRAN Users Reference Manual under the Case Control ELDATA command discussion.

3.5 SEQ text output file

If Bulk data PARAM GRIDSEQ is set to BANDIT, then the SEQGP card images created by the automatic grid sequencer can be output to this file (see Bulk data PARAM entry for parameter GRIDSEQ).

3.6 SPC text output file

If Bulk data PARAM AUTOSPC = Y, then a file containing the SPC1 Bulk Data card images of the SPC's for the AUTOSPC'd DOF can be obtained by using a Bulk Data PARAM entry with a parameter name of PCHSPC1 whose value is Y.

3.7 ANS text output file

This file is generally only useful by the author in the checkout of test problem answers but it is described here for the sake of completeness. It contains all of the answers generated in LINK9, the same as the F06 text file, but none of the F06 info written prior to LINK9. It is only generated if a Bulk Data DEBUG entry is present (see User's Reference Manual for a list of all DEBUG entries).

3.8 PCH text output file

The PCH text file can contain some of the outputs in MYSTRAN (see section 6.3 of the MYSTRAN User's Reference Manual for details of which outputs are allowed)

3.9 L1A text output file

The L1A text file, as well as the Lij binary files (discussed below), are used to communicate data between the LINK's. The L1A file contains data needed to read the binary Lij files. Generally these files are deleted when MYSTRAN completes its execution, but they may be saved via the MYSTRAN initialization file, discussed in a later section

3.10 NEU text output file

The NEU file is created if there is a Bulk Data PARAM entry "PARAM, PRTNEU, YES". This file is used by some commercial software programs for the post-processing of outputs (displacements, forces, stresses, etc) from MYSTRAN. Legacy

3.11 Lij binary output files

In addition to the above mentioned text files that contain output data, MYSTRAN also produces some files that are used to communicate data between the several LINK's of the program. These files all have an extension of three characters length beginning with L followed by a number and a letter (e.g. L2A). The contents of these files, and the details of the way data is written to them, are explained in Appendix A of this manual. Generally these files are deleted when MYSTRAN completes its execution, but they may be saved via the MYSTRAN initialization file, discussed in a later section.

3.12 F2j binary output files

The Case Control ELDATA entry can request output of several items to unformatted binary files in addition to printing of these items to the BUG file. See Appendix A to this manual for the fortran code to read these files.

3 MYSTRAN messages

MYSTRAN writes error, warning and information messages to the F06 file. The error and warning messages are also written to the ERR file.

3.1 Error messages

There are approximately 600 occurrences of more than 300 different fatal error messages that MYSTRAN has. Eventually, these cause MYSTRAN to abort in an orderly fashion with the following written to the F06 and ERR files:

*ERROR nnnn: error message

where nnnn is a 3 or 4 digit error number and error message is a description of what MYSTRAN found to be a fatal error. Sometimes, several fatal error messages will be written prior to MYSTRAN aborting the execution. A specific example of a fatal error message is:

*ERROR 1900: GRID xxxx ON ELEMENT yyyy TYPE type NOT DEFINED

where xxxx is a grid point number and yyyy is an element number and type is the element type (for example BAR).

3.2 Warning messages

There are approximately 60 warning messages that MYSTRAN has. None of these are fatal. The format of these is:

*WARNING : warning message

Warning messages have no message number. An example of the type of warning message that can be written is when MYSTRAN reads an entry from the input data deck but does not recognize the entry. A specific example of a warning message is:

*WARNING : ERROR READING SET ID ON CASE CONTROL SET CARD. CARD IGNORED

3.3 Information messages

There are approximately 100 information messages that MYSTRAN has. The format of these is:

*INFORMATION: information message

Information messages are written to tell various items of interest as MYSTRAN is executing. A specific example of an information message is:

*INFORMATION: GRID POINT xxxx DOF(s)yyyy ARE SINGULAR

where xxxx is a grid point number and yyyy are the displacement components at the grid that are singular.

2.2.2 Changing the default extension for input files

The default extension for input files is dat which can be changed with the command:

DEF EXT ext

where DEF EXT must be in columns 1 through 7 and ext is a character name that must be in column 9 through column 16 and can only consist of 1 to 3 characters and or numbers that are valid for file names. If the extension for a job is not the same as the default extension (either the built in one of dat or one defined with DEF EXT) it must be entered with the file name when MYSTRAN is executed. For example, suppose that you have changed the default extension to D and you want to run EXAMPLE1 that has an extension that is INP. You can either modify the default extension to be INP in the initialization file or enter it with the file name on the command line when MYSTRAN is run (EXAMPLE1.INP).

2.2.3 Setting the level of detail for the log file

The initialization file can also change the level of detail that the log file has. This file contains the start and end times of subroutines. The variable that controls this level of detail is called WRT_LOG and has a default value of 0, which results in no log file at all. Values of WRT_LOG greater than 0 will result in a log file being written. To obtain a log file showing subroutine begin and end times, put the following line into the MYSTRAN.INI file:

WRT_LOG n

where n is an integer > 0. WRT_LOG must begin in column 1 and n can be entered anywhere from column 9 thru column 16. The value n=1 will write begin/end times for the highest level of subroutine calls (LINK's). Successively larger values of n will print more detail. Almost all of the subroutines called in an execution can be shown in the log file by entering a high number for n. A value of n=11 gives about as much detail as can be obtained, but if you want to make sure you get the most detail, enter a higher number (e.g. 99).