

# FortMP Fatal Error Coding List

Last Update  
23 April 2008



INVESTOR IN PEOPLE  
BS EN ISO 9000 : 2000



# FortMP Fatal Error Coding

## FortMP Fatal Errors Etc.

Version: 3.2c

### ***Error Coding - General Introduction***

An error situation in FortMP is handled by issuing an error diagnostic to the screen (or standard output message stream) followed by a 'FATAL ERROR' message. The fatal error message quotes a code number intended to facilitate look-up of the error description. This code number is returned as argument 'TCTN' when FortMP is called as a subsystem, and in the case of MPL-users it is quoted in the status box if the solver exits with an error. For MPL-users the final error-message is also quoted in the status box (latest MPL version).

However this has not yet been systematically implemented and there is no unique coding in existence. Users are advised that each error is identified not by the code in the fatal error message but by the preceding error diagnostic, which must be quoted in any request for user support. User should examine the log-file for more information on how a run has proceeded as this reproduces all messages and diagnostics, and will generally explain what has happened. The log file should be retained as it may be required to enable user support to be given.

A temporary list of error diagnostics, the current associated fatal error code and a brief description (where possible) are given below. This list applies to version 3.2c (July 2000), but is not guaranteed and is not complete. In some cases the error code given may be replaced by another.

### **Warning!**

*Fatal error codes are subject to change without notice*

It is intended to revise the coding with a systematic scheme that will be implemented in successive revisions. Users who have such a revision but do not have the corresponding revision to this document may still find an explanation by searching for the error diagnostic itself and ignoring the fatal error code if it proves misleading.

### ***Fatal Error Code 1 (QP-SSX input error)***

```
Q-FACTORIZE ERROR: <np> -VE PIVOTS FOUND <rn> ROWS NOT PIVOTED
```

Error found when checking that the Q-matrix is positive definite or positive semi-definite. Every pivot found in factorisation should be positive - otherwise it is indefinite.

## FortMP Fatal Error Coding

### ***Fatal Error Code 10***

This is not an error and has no associated error message. It occurs with SPECS commands 'BASREC OFF', 'POSTSOLVE OFF' which cause special output but are very rarely used. Stand-alone FortMP users are not affected. Other users receive this code as return code although it is not an error (this weakness should be corrected in a future revision)

### ***Fatal Error Code 80 (Agenda restriction)***

```
*** AGENDA DATA NOT AVAILABLE FOR LONG NAMES
*** AGENDA DATA NOT AVAILABLE FOR FREE FORMAT
```

Agenda Input feature is not available for input data with long names or in free format.

### ***Fatal Error Code 100 (input error)***

This error code is associated with all (or most) of the errors that can occur in MPS-form input data or basis data. Many such errors are temporarily tolerated until input is complete or the error maximum is reached, and have no final error code of their own. User should see the log-file, which gives details of each fault found. During the first input pass which reads the data and looks up names the system reports any errors together with the input line number and text of the line. Here are some (but not all) of the possible messages that can occur:

```
*** CANNOT OPEN INPUT DATA FILE: <filename>
*** INPUT DATA FILE IS EMPTY
DUPLICATE ROW NAME: <name>
DUPLICATE COLUMN NAME: <name>
CANNOT FIND ROW NAME: <name>
CANNOT FIND COLUMN NAME: <name>
*** TOO MANY INPUT ERRORS
TOO MANY ROWS FOR SHORT INDEXING
NO OF ROWS EXCEEDS LICENSED MAXIMUM
NO OF COLUMNS EXCEEDS LICENSED MAXIMUM
PREMATURE END OF INPUT FILE
ERROR IN DECODING THE INPUT DATA
ERROR IN READ FROM THE INPUT DATA
ORG-TYPE MARKER REPEATS WITHOUT END BETWEEN
INTEND MARKER DOES NOT FOLLOW INTORG
TYPE OF SOS NOT RECOGNISED
SETEND MARKER DOES NOT FOLLOW SETORG
CANNOT OPEN BASIS INPUT
FATAL ERROR - EXECUTION HALTED
TOO MANY ERRORS - EXECUTION HALTED
```

## FortMP Fatal Error Coding

### **Fatal Error Code 100 (external interface error)**

UNRECOGNISED CODE IN STATUS TABLE BSTAT

This error indicates that the BSTAT-array, used to enter a starting basis in SUBMP1 calls, contains wrongly coded entries. Entries should be: Zero for basic, +1 for non-basic at bound and -1 for non-basic at zero.

### **Fatal Error Code 100 (QP-IPM error)**

\*\*\* NOT ENOUGH MEMORY FOR SYMBOLIC FACTOR \*\*\*\*

Indicates that the normal (Cholesky) factorisation requires too much memory. This may be expected if the model has one or more columns with high density. The present IPM system cannot solve the QP problem in this case.

### **Fatal Error Code 100 (QP-IPM error)**

```
*** IMPROPER VALUE : DIVISION BY ZERO AVOIDED ***
*** ARRAY WORK SPACE FINISHED *****
***ABNORMAL PROGRAM TERMINATION, PROGRAM HALTED***
*** UNEXPECTED COLUMN TYPE FROM FORTCV *****
*** UNEXPECTED ROW TYPE FROM FORTCV *****
*** INPUT PROBLEM TO BIG , FATAL ERROR *****
*** COLUMN INDEX OUT OF RANGE ,FATAL ERROR *****
*** SINGULARITY DETECTED IN SYMMETRIC MATRIX *****
*** ILLEGAL VALUE IN SOLUTION DETECTED *****
*** SYMMETRIC MATRIX IS NOT CONNECTED *****
*** LOGICAL PROGRAMMING MISTAKE ****
*** NOT ENOUGH MEMORY FOR SYMMETRIC MATRIX ****
*** NOT ENOUGH SPACE IN WORK ARRAY ****
*** Zero subscript in the array, program halted ***
*** TRIVIAL PROBLEM:- ZERO ROWS *****
```

The above errors result from checks made during the IPM algorithm (QP models). Mostly these errors should not occur, and user should refer to user support.

### **Fatal Error Code 100 (QP-IPM error)**

```
*** Unknown error number : 20****
***IPMABC: OFFENDING SUBROUTINE : MNDGAD ***
```

The above errors result from the algorithm that builds pre-determined sequence for the Cholesky. It can occur if the QP data has gaps - for example a constraint-row that is entirely empty (or only has coefficients on fixed columns).

## FortMP Fatal Error Coding

### ***Fatal Error Code 130 (input error)***

The second input pass handles not only input MPS-form data, but also the external interface data that is passed as arguments when FortMP is called as a subsystem, or is invoked by MPL or AMPL.

```
NEGATIVE RANGE ON ROW# <row>  
UPBND < LOBND ON COL# <col>
```

The above diagnostics refer to inconsistent bounds or ranges.

```
SEMI-CONTINUOUS ERROR: UPBND < 1.0, COL# <col>
```

The above diagnostic refers to a semi-continuous variable - type SC. An SC-type variable has its continuous range bounded below by 1.0.

### ***Fatal Error Code 150 (input error)***

Second input pass, or sub-system input data error. See also code 130.

```
GAPS OR OTHER ERROR IN MATRIX STRUCTURE  
THE LAST PACKED COLUMN IS NOT NCOL
```

The above diagnostic refers to the input matrix data. There must be at least one entry (non-zero) for every column.

### ***Return Error Code 150 (SPCDFT error)***

```
*** FAULT IN THE VALUE, DEFAULT IS USED ***  
*** FAULT IN 1ST KEYWORD, COMMAND NOT ACCEPTED ***  
*** FAULT IN 2ND KEYWORD, COMMAND NOT ACCEPTED ***  
*** FAULT IN 3RD KEYWORD, COMMAND NOT ACCEPTED ***  
*** FAULT IN 4TH KEYWORD, COMMAND NOT ACCEPTED ***
```

System-users may call SPCDFT prior to the main library entries (SUBMP1 and SUBQP1) in order to change the default value for a command. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above messages refer to invalid commands with the faulty command listed on the previous line.

## FortMP Fatal Error Coding

### ***Fatal Error Code 160 (input error)***

Second input pass, or sub-system input data error. See also code 130.

```
BOUNDS REVERSED OR NEGATIVE RANGE - FATAL
```

The above diagnostic refers input data for bounds or ranges.

### ***Fatal Error Code 170 (input error)***

Second input pass, or sub-system input data error. See also code 130.

```
TOO MANY ROWS FOR SHORT INDEXING
```

The above diagnostic refers to the input data. It occurs only in custom 'Short Index' versions of the system when the model has more than 32K rows (with cuts and other extras included)

### ***Fatal Error Code 175 (licensing error)***

This code indicates a licensing error. The user may not be licensed at all, may not have the license file installed correctly, or may simply have a license that does not cover the feature invoked by his run. Associated error diagnostics are as follows:

```
ASSIGN <arrname> , LICENSED MAXIMUM EXCEEDED  
NO OF ROWS EXCEEDS LICENSED MAXIMUM  
NO OF COLUMNS EXCEEDS LICENSED MAXIMUM  
IPM IS NOT LICENSED  
MIP IS NOT LICENSED  
QP-IPM IS NOT LICENSED  
INTEGER QP IS NOT LICENSED  
QP-SSX IS NOT LICENSED  
QP-SSX-MIP IS NOT LICENSED
```

### ***Fatal Error Code 187 (QP-SSX input error)***

```
DIAGONAL ENTRY MISSING IN Q-MATRIX
```

Error found when checking that the Q-matrix is positive definite or positive semi-definite. Every row (column) of the symmetric Q-matrix should have a diagonal entry - otherwise it is indefinite.

## FortMP Fatal Error Coding

### ***Fatal Error Code 188 (licensing error)***

DEMO ONLY:- PRIMAL SOLUTION NOT RECORDED

This error code is issued when user is not licensed for output. The primal output solution contains only zero values. It should occur only in special versions, as output is pre-licensed and not subject to control in the standard release.

### ***Fatal Error Code 188 (QP-SSX input error)***

Q-COLUMN ENTRY FOUND OUTSIDE THE Q-ROWS

Error found when checking that the Q-matrix is positive definite or positive semi-definite. Every row (column) of the symmetric Q-matrix must have a diagonal entry - otherwise it is indefinite. A diagonal entry means the Q-row must exist, so if it does not exist then Q is indefinite.

### ***Fatal Error Code 190 (input error)***

QMATRIX INPUT IS NOT CONSISTENT - sbr INFQT1

The above diagnostic refers to the input QP data received as arguments by 'SUBQP1' (that is received from MPL).

### ***Fatal Error Code 199 (user cancel)***

Associated error diagnostics are as follows:

RUN IS CANCELLED  
RUN CANCELLED FROM MPL  
QP-SSX RUN IS CANCELLED FROM MPL

The above error diagnostics indicate that the run has been cancelled by the user. They may occur, for example when an MPL-user clicks the 'Cancel' button on the MPL status box.

### ***Fatal Error Code 199 (input error)***

CANNOT MAXIMIZE F-ROW FORM QP

QP data introduced by means of F-data cannot be handled if the objective sense is 'MAXIMIZE'.

## FortMP Fatal Error Coding

### ***Fatal Error Code 200 (iteration limit)***

```
ITERATION LIMIT REACHED - USE RESTART TO CONTINUE  
ITERATION LIMIT EXCEEDED
```

This code indicates that the iteration limit for an SSX algorithm has been reached. Provided restart-points have been saved the run may be continued with appropriate 'RESTART' commands.

### ***Fatal Error Code 200 (file error)***

```
READ ERROR ON CHANNEL NO <ichan>
```

Indicates a formatting or other error on an input file. See User Support for more details

### ***Fatal Error Code 280 (QP-SSX-MIP restart error)***

```
AGENDA BASIS STATUS ERROR
```

These errors indicate something wrong in the '.AGL' file used to restart an integer solution in QP-SSX MIP algorithm. Either the file has been corrupted or restart is attempted with a model having a different structure.

### ***Fatal Error Code 300 (IPM error)***

```
*** NOT ENOUGH MEMORY FOR SYMBOLIC FACTOR ***
```

Indicates that the normal (Cholesky) factorisation requires too much memory. This may be expected if the model has one or more columns with high density. The present IPM system cannot solve the problem in this case.

### ***Fatal Error Code 300 (IPM error)***

```
*** CHOLESKY ERROR NORM EXCEEDS TOLERANCE ***
```

This error indicates numerical difficulty with equation solving due to an ill-conditioned Cholesky. Attempt can be made to solve the problem by increasing 'CHOLESKY ERROR TOLERANCE', but in any case it has been found unlikely that the iterations will converge to a solution in this case. The error may be caused by an infeasible or unbounded model.

## FortMP Fatal Error Coding

### **Fatal Error Code 300 (IPM error)**

```
***  MODEL IS PRIMAL UNBOUNDED      *****
***  MODEL IS DUAL UNBOUNDED        *****
```

These situations are normal and should result in normal exit with status 1 (infeasible) or 2 (unbounded). Refer to user support if appearing with fatal error code 300.

### **Fatal Error Code 300 (IPM error)**

```
***  IMPROPER VALUE : DIVISION BY ZERO AVOIDED ***
***  ARRAY WORK SPACE FINISHED *****
***  ABNORMAL PROGRAM TERMINATION, PROGRAM HALTED ***
***  UNEXPECTED COLUMN TYPE FROM FORTCV *****
***  UNEXPECTED ROW TYPE FROM FORTCV *****
***  INPUT PROBLEM TO BIG , FATAL ERROR *****
***  COLUMN INDEX OUT OF RANGE , FATAL ERROR *****
***  SINGULARITY DETECTED IN SYMMETRIC MATRIX *****
***  ILLEGAL VALUE IN SOLUTION DETECTED *****
***  SYMMETRIC MATRIX IS NOT CONNECTED *****
***  LOGICAL PROGRAMMING MISTAKE ****
***  NOT ENOUGH MEMORY FOR SYMMETRIC MATRIX ****
***  NOT ENOUGH SPACE IN WORK ARRAY ****
***  Zero subscript in the array, program halted ***
***  TRIVIAL PROBLEM:- ZERO ROWS *****
```

The above errors result from checks made during the IPM algorithm (LP and QP models). Mostly these errors should not occur, and user should refer to user support.

### **Return Error Code 300 (SPCINT error)**

```
SPECS-error:- Not terminated by END
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

## FortMP Fatal Error Coding

### ***Return Error Code 310 (SPCINT error)***

```
SPECS-error:- Nothing found
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

### ***Return Error Code 320 (SPCINT error)***

```
SPECS-error:- Cannot open fortmp.spc
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

### ***Return Error Code 330 (SPCINT error)***

```
SPECS-error:- Too many BEGIN line
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

### ***Return Error Code 340 (SPCINT error)***

```
SPECS-error:- Too many commands
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

## FortMP Fatal Error Coding

### **Return Error Code 350 (SPCINT error)**

```
SPECS-error:- Too much text
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above is a possible error.

### **Fatal Error Code 380 (QP-SSX restart error)**

```
BASIS FOUND NON-COMPLEMENTARY AFTER B-BASIS RESTART
```

A binary basis used to restart QP-SSX with the command 'SSX START RESTART' must be complementary and should be the optimum from a previous run.

### **Return Error Code 390 (SPCINT error)**

```
*** FAULT IN THE VALUE, DEFAULT IS USED ***
*** FAULT IN 1ST KEYWORD, COMMAND NOT ACCEPTED ***
*** FAULT IN 2ND KEYWORD, COMMAND NOT ACCEPTED ***
*** FAULT IN 3RD KEYWORD, COMMAND NOT ACCEPTED ***
*** FAULT IN 4TH KEYWORD, COMMAND NOT ACCEPTED ***
```

System-users may call SPCINT prior to the main library entries (SUBMP1 and SUBQP1) in order to load the SPECS command file 'FORTMP.SPC' once only and save time on repeated entries which must then refer to the SPECS internally. Errors are not recorded on the log (as no log is as yet defined) but are written to screen output. The above messages refer to invalid commands with the faulty command listed on the previous line.

### **Fatal Error Code 500 (infeasibility)**

Apart from restart-errors given below, this code is reserved for Primal infeasibility detected by the PRESOLVE algorithm, or by MIP preprocessor. Some of the associated error diagnostics are:

```
PROBLEM FOUND INFEASIBLE (SNGROW)
PROBLEM FOUND INFEASIBLE BY PRESOLVE (REDROW)
PROBLEM FOUND INFEASIBLE BY PRESOLVE (REMBND)
```

Additional details of the infeasibility found are given in the log.

## FortMP Fatal Error Coding

### ***Fatal Error Code 500 (restart error)***

Other errors with this code are:

```
SAVE-FILE INCOMPATIBLE - CANNOT RESTART INPUT
SAVE-FILE ABSENT OR READ-ERROR - CANNOT RESTART INPUT
CANNOT OPEN MATRIX FILE FOR RESTART
CANNOT OPEN NAMES FILE FOR RESTART
```

The above diagnostics indicate that it was impossible to do an 'INPUT' restart owing to some error with previously saved data.

### ***Fatal Error Code 550 (unbounded)***

This code is reserved for Dual infeasibility detected by the PRESOLVE algorithm. Some of the associated error diagnostics are:

```
DUAL PROBLEM FOUND INFEASIBLE (SFRCOL)
DUAL PROBLEM FOUND INFEASIBLE (SNGCOL)
CANNOT FIX VARIABLE - PROBLEM FOUND UNBOUNDED (FIXCOL)
```

Dual infeasibility normally implies that the primal model is unbounded, but the primal model may also be infeasible. Additional details of the unbounded condition (or infeasibility) are given in the log.

### ***Fatal Error Code 700 (memory overflow)***

```
NOT ENOUGH MEMORY-SPACE FOR THE PROBLEM
ASSIGN <arrname> , MEMORY IS NOT AVAILABLE
MEMORY ALLOC FAILED IN IPM
```

The above errors indicate failure to find memory-space in a region already allocated to hold problem data and temporary stores. The algorithm should pre-calculate the space needed so that this error does not occur. However storage available may be restricted by the licensing and if this is so a warning message about the licensed storage restriction will have appeared.

In the case of QP, storage pre-calculation must be assisted by giving an 'extension' that the user can control, and this may be necessary for very large models. Use one of the equivalent SPECS commands:

```
MEMORY EXTENSION = n
SET QPMEMEXTENSION = n
```

to enlarge the space allocation. Default for 'n' is 10000000 (10 million bytes).

## FortMP Fatal Error Coding

### ***Fatal Error Code 700 (memory overflow)***

```
NOT ENOUGH SPACE FOR ETA-FILE - sbr MFORMU  
NOT ENOUGH SPACE FOR ETA-FILE - sbr MFORML
```

The above errors indicate memory problems during re-inversion. Refer to User Support.

### ***Fatal Error Code 700 (memory overflow)***

```
NOT ENOUGH MEMORY FOR U-MATRIX IN IPM
```

The above error indicates that U-factor building ran out of space in QP-IPM. Should not occur - refer to user support.

### ***Fatal Error Code 733 (memory overflow)***

```
NOT ENOUGH SPACE FOR MARKOWITZ/ETA ASSIGNMENT
```

The above error indicates a memory problem in assigning space for basis inversion. It is not expected to occur - refer to User Support.

### ***Fatal Error Code 750 (memory overflow)***

```
NOT ENOUGH SPACE FOR U-ETA COLUMN TRANSPOSE
```

The above error indicates a memory problem in preparing the U-eta file. It is not expected to occur - refer to User Support.

### ***Fatal Error Code 750 (memory overflow)***

```
*** CANNOT ALLOCATE ENOUGH MEMORY FOR THE TREE
```

The above error indicates there is insufficient memory to run the MIP algorithm. MIP will proceed so long as it is possible to allocate space for at least one tenth of the nodes requested (and FIX and CUT stores to match according to quota). Failure may indicate that memory is overloaded and the run may succeed if some other tasks are ended. Otherwise enter a 'MAX MIP SPACE' command with a reduced node count (default is 10000).

## FortMP Fatal Error Coding

### ***Fatal Error Code 750 (memory overflow)***

Revised dimensions are returned

This diagnostic is returned and this code is issued when returning from a system-users call to subroutine 'MP1INP' if there is not enough space in the arrays passed to receive the input problem. User should increase the size allocated before repeating and the necessary dimension-sizes are returned in the corresponding arguments. Details are to be found in the log.

### ***Fatal Error Code 776 (memory overflow)***

CANNOT FIND ETA-SPACE AFTER INVERT - ABORTED

The above error diagnostic indicates that no space remains in the eta-file for SSX iterations after inversion. It may occur in any variant of SSX - Primal, Dual, Basis Recovery or QP-SSX. Refer to user support for assistance in overcoming the problem.

### ***Fatal Error Code 780 (Cut processing error)***

MAXIMUM CUT NON-ZEROS EXCEEDED  
MAXIMUM CUT ROWS EXCEEDED

Not enough space is available in the Cut Pool. This may be overcome by increasing the Cut Quota.

### ***Fatal Error Code 780 (Cut processing error)***

PUTCUT COLS OUT OF ORDER

This error is not expected to occur. Refer to User Support.

### ***Fatal Error Code 790 (memory overflow)***

ASSIGN <arrname> , MEMORY IS NOT AVAILABLE

The above error indicates failure to allocate memory-space to hold temporary stores. This means that the system has not been able to allocate space because virtual memory is full. Some other systems need to be cancelled in order to run the problem

## FortMP Fatal Error Coding

### ***Fatal Error Code 799 (memory overflow)***

NOT ENOUGH ROOM FOR BASIS-INVERT

The above error indicates a memory problem in initial loading of the basis for inversion. It is not expected to occur - refer to User Support.

### ***Fatal Error Code 799 (memory overflow)***

NOT ENOUGH SPACE FOR BASIS UPDATE

The above error indicates a memory problem in updating the basis sub-bump during inversion. It is not expected to occur - refer to User Support.

### ***Fatal Error Code 799 (memory overflow)***

NOT ENOUGH SPACE FOR MKW/ETA ASSIGNMENT - sbr BCRASH

The above error indicates a memory problem in the 'CRASH' stage of basis recovery (or crossover). Refer to User Support.

### ***Fatal Error Code 799 (memory overflow)***

NOT ENOUGH SPACE FOR SYMMETRIC BASIS UPDATE

The above error indicates a memory problem in updating the symmetric matrix for QP-SSX positive semi-definite check. Refer to User Support.

### ***Fatal Error Code 800 (restart error)***

OPEN OR READ ERROR ON SSX RESTART FILE  
OPEN OR READ ERROR ON BINARY BASIS INPUT FILE

The above errors occur when 'SSX RESTART' is attempted without the proper restart file.

## FortMP Fatal Error Coding

### ***Fatal Error Code 800 (file error)***

```
CHANNEL <n> ALREADY IN USE - CANNOT BE RE-OPENED
```

The above error occurs if one of the local FORTRAN I/O units required by the system is pre-empted. FortMP may use unit numbers 16 to 28 and these units should be avoided by any FORTRAN-coded system that calls the subroutine library.

```
OPEN ERROR FILE: <fnam>
```

This error indicates failure to open an input file (data file, save file or internal file). Usually there is another following error diagnostic. Refer to User support if not.

### ***Fatal Error Code 800 (internal interface error)***

```
CVTAB/IRTAB DIMENSION EXCEEDED IN GETCOL  
RVTAB/JCTAB DIMENSION EXCEEDED IN GETROW  
MAXIMUM EXTRA COLUMNS EXCEEDED  
DUPLICATE ROW IN ADDCOL  
MAXIMUM EXTRA ROWS EXCEEDED  
UP NOT > LO WITH TYPE BOUND - SBR CHGRIM  
UNEQUAL BOUNDS WITH TYPE FIXED - SBR CHGRIM  
UP NOT > LO WITH TYPE BOUND - SBR CHGCTP  
UNEQUAL BOUNDS WITH TYPE FIXED - SBR CHGCTP  
UP NOT > LO WITH TYPE BOUND - SBR CHGRTP  
UNEQUAL BOUNDS WITH TYPE FIXED - SBR CHGRTP
```

The above errors can occur when system-calls are made employing the Internal Interface (see manual chapter 8)

### ***Fatal Error Code 800 (Row Classify error)***

```
rvtab/jctab Dimension exceeded in GETROW
```

Work-area size is too small in the row classify routine. Should not occur - see User Support.

### ***Fatal Error Code 800 (phase 1 error)***

```
CANNOT FIND PIVOT IN PHASE I - ABORTED
```

The above error results in the Primal algorithm Phase I (infeasible) when the ratio-test cannot find a pivot. In Phase II this would indicate that the model is unbounded but in Phase I it cannot theoretically occur. Refer to User Support.

## FortMP Fatal Error Coding

### ***Fatal Error Code 800 (Cut processing error)***

MAXIMUM EXTRA ROWS EXCEEDED

The number of cuts to be generated exceeds the allowance for extra rows. It can be overcome by assigning an enlarged quota.

### ***Fatal Error Code 800 (Cut processing error)***

MAX COLSPACE EXCEEDED - sbr APLCT1

The number of cuts to be generated exceeds the allowance for extra column-space. It can be overcome by assigning an enlarged quota.

### ***Fatal Error Code 800 (QP-SSX error)***

Q-MATRIX FOUND INDEFINITE AFTER INVERT

This message indicates that QP-SSX was unable to complete execution owing to the Q-matrix being indefinite (not convex). However it may also occur as a result of numerical trouble that the code was unable to handle. Refer to User Support

### ***Fatal Error Code 800 (QP-SSX error)***

DUAL VAR NON-BASIC BUT NOT AT ZERO  
PRIMAL VAR HAS ILLEGAL TYPE  
COMPLEMENTARY BASIC PAIR  
IE= <ie>, J1/ST/TP= <J1> <S1> <T1>, J2/ST/TP= <J2> <S2> <T2>

Error in basis 'rectification' when restarting a sub-problem. Not expected to occur - refer to User Support.

### ***Fatal Error Code 801 (QP-SSX error)***

STATUS = 0 - PRIMAL-UNBOUNDED BUT NOT DUAL-FEASIBLE

The above error may occur in QP-SSX and is not recoverable. Theoretically it should be impossible. Refer to User Support

## FortMP Fatal Error Coding

### ***Fatal Error Code 850 (Dual error)***

```
850 DUAL INFEASIBLE - NOT TOLERATED
```

Error returned by the Dual algorithm running only in Phase II and with dual infeasibility greater than tolerance. Normally Dual can run in both Phase I and Phase II but user has the option to run only in Phase II with the command 'DUAL PHONE OFF' (default is 'ON'). The situation is normally recovered by reverting to the Primal algorithm.

### ***Fatal Error Code 851 (Dual error)***

```
DUAL INFEASIBILITY REMAINS - ABNORMAL EXIT
```

A check on dual feasibility is made before final exit to ensure that the exit condition can be reliably used by the MIP algorithm. If the check fails the situation is normally recovered by reverting to the Primal algorithm.

### ***Fatal Error Code 870 (memory overflow)***

```
WRITE ETA PROBLEM IN PRIMAL - CANNOT RECOVER  
WRITE ETA PROBLEM IN DUAL - CANNOT RECOVER  
WRTETA PROBLEM:- FATAL ERROR
```

This error follows an unrecoverable memory problem with the ETA-file. See also error code 776 above as one those diagnostics should precede this one.

### ***Fatal Error Code 876 (numerical difficulty)***

```
ALL CANDIDATES FLAGGED AFTER INVERT - ABORTED  
ALL CANDIDATES FLAGGED AFTER INVERT - DUAL ABORT  
ALL CANDIDATES FLAGGED AFTER INVERT
```

The above diagnostics are issued from the SSX and QP-SSX pivoting when all possible candidates are ruled out because their pivot-sizes are too small to be reliable. In the Dual algorithm the problem may still be solved by reverting to the Primal algorithm.

## FortMP Fatal Error Coding

### ***Fatal Error Code 876 (numerical difficulty)***

```
PIVOT ROW NOT FOUND IN COLUMN - sbr WRTETA  
ZERO PIVOT ELEMENT IN WRTETA
```

These errors can possibly occur as a result of extreme numerical trouble, and in any case are normally recovered by re-inverting.

### ***Fatal Error Code 877 (QP-SSX error)***

```
*** BACKOUT LIST OVERFLOW. CANNOT RECOVER  
*** BACKOUT FLAG LIST OVERFLOW. CANNOT RECOVER
```

This error is issued by QP-SSX solver when a pivot must be flagged but it is impossible to do so because there is no space to save the necessary data. Flagging in the case of QP-SSX involves 'Backout', that is to say undoing all minor iterative steps since the most recent major step taken with the basis in standard (complementary) form. The error can only occur with a quite exceptionally long series of minor iterations.

### ***Fatal error Code 880 (QP-SSX error)***

```
SINGULAR BASIS FOUND TOO MAY TIMES
```

Singularity of the basis at invert time in QP-SSX is recovered by backing out and re-inverting. The process can produce a cycle and infinite cycling is preventing by counting to a maximum the number of permitted times it can occur.

### ***Fatal Error Code 899 (QP-SSX input error)***

```
BAD MATRIX SORT OR MISSING COLUMN - sbr BPDMAT
```

This error occurs if the matrix input in a QP problem has an empty constraint row other than the objective row. When the primal-dual matrix is constructed this creates an empty column which is not handled by the current version.

## FortMP Fatal Error Coding

### ***Fatal Error Codes 899 to 999***

One case of code 899 is described above - it will be renumbered in the next revision. These codes are reserved for software checks and, if they occur, it normally indicates an illogical situation. A large number of these checks are made and the associated error messages cannot be reproduced here. In the event of such an error (other than the one above) user is requested to save his problem data and send it to User Support for bug checking together with the log file.

For MPL-users, AMPL-users and users of systems that call the FortMP subroutine library it is probably inconvenient to send all the information needed to reproduce the problem. However this is not necessary. All of the data passed over the External Interface when a call is made to SUBMP1 or SUBQP1, or when MPL/AMPL invokes the FortMP solver, is saved in a binary file which allows the problem to be reproduced without the users external system. This feature is invoked with the command:

```
EXTERNAL SAVE ON
```

It causes a file named '<model>.XFC' to be created in the working directory (together with the log) and this can be sent instead of the whole system.