THE DESIGN AND USAGE OF THE NEW DATA MANAGEMENT FEATURES IN NASTRAN®

P. R. Pamidi and W. K. Brown
RPK Corporation

SUMMARY

Two new data management features have been installed in the April 1984 release of NASTRAN. These two features are the Rigid Format Data Base and the "READFILE" capability. The Rigid Format Data Base is stored on external files in card-image format and can be easily maintained and expanded by the use of standard text editors. This data base provides the user and the NASTRAN maintenance contractor with an easy means for making changes to a Rigid Format or for generating new Rigid Formats without unnecessary compilations and link editing of NASTRAN. Each Rigid Format entry in the data base contains the Direct Matrix Abstraction Program (DMAP), along with the associated restart, DMAP sequence subset and substructure control flags. NASTRAN reads a specific entry in the data base directly in every NASTRAN run and performs the necessary transformations to allow the DMAP to be processed and compiled by the NASTRAN executive.

The "READFILE" capability allows an user to reference an external secondary file from the NASTRAN primary input file and to read data from this secondary file. There is no limit to the number of external secondary files that may be referenced and read. The "READFILE" capability may be invoked anywhere in the Executive Control, Substructure Control, Case Control and Bulk Data Decks.

INTRODUCTION

The April 1984 release of NASTRAN has been enhanced by the addition of two new features. These are the Rigid Format Data Base and the "READFILE" capability. Both of these features greatly add to the flexibility and versatility of NASTRAN. They are described in detail in the following sections.

RIGID FORMAT DATA BASE

The new Rigid Format Data Base allows for convenient maintenance of the existing Rigid Formats and the addition of new Rigid Formats. Editing of the data base may be done by using standard text editors provided on the host computer systems. The basic rationale and the advantages of the data base have been discussed in an earlier paper (Reference 1).

Design of the Data Base

The Rigid Format Data Base is a collection of all Rigid Formats available to the user in NASTRAN. Each Rigid Format is maintained as a separate card-image entry within the data base. The entry for each Rigid Format consists of three parts. The first part is the DMAP part. It contains the DMAP sequence for the Rigid Format, the DMAP sequence subset flags, the restart flags (card name, file name and Rigid Format switch restart flags) and the substructure DMAP ALTER control flags. The second part contains the card name table and the third part contains the file name table. The restart flags in the first part and the name tables comprising the second and third parts are not processed by NASTRAN in non-restart runs. Similarly, the substructure control flags in the first part are not processed in non-substructure runs.

The format of the data base is free field. Each of the three parts in a Rigid Format entry is separated from the other parts by a "$*" card. The following fictitious example illustrates a Rigid Format entry in the data base.

```
APR.84
$$$$ THIS IS A COMMENT
$$$$ ******************************************************
MODULE1    IN1,IN2,/OUT1,OUT2//*PARM1* $
****SBST   1,3,9-12
****RFMT   188,200-201
****CARD   1-20,30,44
****FILE   100-104,110
****PHS1   I1
****PHS2   DB5
****PHS3   D7
$$$$ ******************************************************
MODULE2    IN3,IN4/OUT3/*PARM2* $
****CARD   1-40,45
****FILE   101,102
****PHS2   DE5
$$$$ ******************************************************
```

```
          .
          .
          .
$$$$
$*CARD NAME TABLE
$$$$
1         AXIC    AXIF    CELAS1    CELAS2
2         ADUM1   CDUM1   CROD
          .
          .
          .
$$$$
$*FILE NAME TABLE
$$$$
94        SLT     GPTT
95        KGGX    GPST
          .
          .
          .
$*
```

The very first card of an entry identifies the release of NASTRAN with which the Rigid Format is associated. In this example, the Rigid Format is associated with the April 1984 release.

The "$*CARD" card separates the card name table from the DMAP part of the entry and the "$*FILE" card separates the file name table from the card name table. A "$*" card terminates the file name table and the Rigid Format entry.

Comment cards are identified in the data base by the "$$$$" identification in the first four columns of the field and control cards are identified by the "****" identification in the first four columns of the field.

Comment cards may be placed anywhere in the card name or file name tables (the second and third parts of a Rigid Format entry). However, comment cards have a required usage and serve a specific purpose in the DMAP part of a Rigid Format entry. In this part, a comment card is used to distinguish and separate a DMAP entry (that is, a DMAP statement and its associated control cards) from another DMAP entry. Hence, there must be at least one comment card separating a DMAP entry from the next DMAP entry. In the data base supplied with NASTRAN, a comment card with a trailing string of "*" is used for this purpose to serve as a cosmetic delineation between successive DMAP entries.

All DMAP statements must conform to the rules as specified in the NASTRAN User's Manual (Reference 2). Any card in the DMAP part of a Rigid Format entry that does not begin with "$$$$" or "****" in the first four columns of the field is considered to be a DMAP statement or part of a DMAP statement.

Comment and control cards in a Rigid Format entry can extend up to 80 columns. However, DMAP cards can only extend up to 72 columns.

Control cards (that is, cards that begin with "****" in the first four columns of the field) are permitted only in the DMAP part of a Rigid Format entry. A control card must have any one of seven four-character names in columns five through eight. The permissible names are: SBST, RFMT, CARD, FILE, PHS1, PHS2 and PHS3. Control cards follow the corresponding DMAP statement in the entry and may be specified in any order.

The "SBST", "RFMT", "CARD" and "FILE" control cards contain sequences of numbers and/or ranges of numbers in ascending order represented by the use of a dash. A comma is required after each number in a sequence or after a range of numbers, if an additional number or range of numbers is to follow. There may be multiple cards for any one of these control cards for a specific DMAP statement.

The "SBST" control card provides DMAP sequence subset controls. If a user requests a given subset on the "SOL" card of a NASTRAN run and that number is in the sequence of numbers given on the "SBST" card, then the associated DMAP statement is deleted. The range of subset numbers is from 1 to 9 and each number is documented in the NASTRAN User's Manual (Reference 2).

The "RFMT" control card is processed in restart runs and is applicable to cases where a Rigid Format switch has occurred. Each Rigid Format has a unique number assigned to it. For APPROACH DISP, Rigid Formats 1 through 15 are assigned numbers 187 through 201. For APPROACH HEAT, Rigid Formats 1, 3 and 9 are assigned numbers 207, 208 and 209. For APPROACH AERO, Rigid Formats 10 and 11 are assigned numbers 214 and 215. A DMAP statement is flagged for execution in a modified restart if the number associated with the Rigid Format that was used in the checkpointed run is listed in the sequence of numbers given on the "RFMT" card provided with the DMAP statement.

The "CARD" and "FILE" control cards provide restart information for changes that involve input data or files within the DMAP. For a given Rigid Format, every type of effective change in the Case Control and Bulk Data Decks and each output file (or data block) in the DMAP is assigned a number as defined in the card name and file name tables in the second and third parts of a Rigid Format entry. In a modified restart, if the number associated with an input data change or an affected file appears in the sequence of numbers given on the "CARD" or "FILE" cards, then the corresponding DMAP statement is flagged for execution in the restart run.

The "PHS1", "PHS2" and "PHS3" control cards are used to indicate where substructure DMAP ALTERs are to be generated. The number following the "PHS" refers to the substructure phase

number. These cards must have one of the following flags: "In", "Dn", "DBn" or "DEn". The "n" in these flags is an integer that refers to the subroutine governing the substructure run (subroutine ASCM01, ASCM05, ASCM07 or ASCM08) and must have the value "1" for Phase 1 cards, either the value "5" or "8" for Phase 2 cards, and either the value "1" or "7" for Phase 3 cards. The "I" in the "In" flag indicates that a DMAP ALTER is to be inserted after this DMAP statement. The "D" in the "Dn" flag indicates that this DMAP statement is to be deleted and possibly replaced by a DMAP ALTER. The "DB" in the "DBn" flag and the "DE" in the "DEn" flag indicate the beginning and the end of a group of contiguous DMAP statements that are to be deleted and possibly replaced by a DMAP ALTER. Users are cautioned to be very careful in making any changes to these substructure control cards because of their impact on the DMAP ALTERs automatically generated in substructure analyses. (The substructure capability is currently implemented only in Rigid Formats 1, 2, 3, 8 and 9, APPROACH DISP.)

The card name and file name tables assign numbers to every type of effective change in the Case Control and Bulk Data Decks and to every output file (or data block) in the DMAP. Numbers 1 through 93 are assigned to card names and numbers 94 through 186 are assigned to file names. This information is used subsequently to determine the DMAP statements to be flagged for execution in modified restarts. The format of these tables is free field. Each entry in these tables must have an integer number in the first field and a list of names in the remaining fields of the entry. All names are to be alphanumeric and may contain up to a maximum of eight characters. No name should appear twice in these tables. Comment cards may be freely used in these tables to facilitate readability.

Implementation of the Data Base

The Rigid Format Data Base is implemented differently on the CDC, DEC VAX, IBM and UNIVAC versions. On the CDC and DEC VAX versions, each Rigid Format entry is stored as a separate file. The local names of these files during a NASTRAN execution are: DISP1 through DISP15 for APPROACH DISP; HEAT1, HEAT3 and HEAT9 for APPROACH HEAT; and AERO10 and AERO11 for APPROACH AERO. These same files are stored as members of a partitioned data set (PDS) on the IBM version and as elements of the *NASTRAN file on the UNIVAC version. The member and element names are exactly the same as the local file names on the CDC and DEC VAX versions. On the IBM version, the PDS containing the Rigid Format Data Base must be referred to by a Data Definition card, "DD", with the DDname of RFDATA. On the UNIVAC version, the *NASTRAN file is the file containing the NASTRAN program absolutes. (See References 3 and 4 for the formats of file names for the CDC and DEC VAX versions, respectively. See Reference 5 for the formats of DDnames and member names for the IBM version. See Reference 6 for the format of UNIVAC file names.)

Usage of the Data Base

The following examples illustrate the manner in which the Rigid Format Data Base is accessed and used on all of the four versions of NASTRAN.

CDC VERSION

```
/JOB.
.
.
GET,DISP1,DISP2,DISP3,DISP4,DISP5.
GET,DISP6,DISP7,DISP8,DISP9,DISP10.
GET,DISP11,DISP12,DISP13,DISP14,DISP15.
GET,HEAT1,HEAT3,HEAT9,AERO10,AERO11.
RFL,220000.
REDUCE,-.
LINK1,INPUT,OUTPUT,PUNCH,UT1.
/EOR
ID ....
.
.
ENDDATA
/EOF
```

DEC VAX VERSION

```
ASSIGN  DDB1:[NASDIR]DISP1.DT DISP1.
ASSIGN  DDB1:[NASDIR]DISP2.DT DISP2.
        .
        .
ASSIGN  DDB1:[NASDIR]HEAT1.DT HEAT1.
        .
        .
ASSIGN  DDB1:[NASDIR]AERO11.DT AERO11.
@DDB1:[NASDIR]NASTRAN   DEMO.DT
```

IBM VERSION

```
//   EXEC   NASTRAN
//NS.RFDATA  DD  DSN=RIGID.FORMAT.DATA,DISP=SHR
//NS.SYSIN   DD  *
ID ....
.
.
ENDDATA
//
```

UNIVAC VERSION

```
@ASG,A    *NASTRAN.
@XQT      *NASTRAN.LINK1
```

Development of User Rigid Formats

In addition to using COSMIC-supplied Rigid Formats, users may develop their own Rigid Formats, with restart capabilities included. Rigid Formats developed by users must conform to the rules explained earlier and must be similar in content and structure to the COSMIC-supplied Rigid Formats. Each user-developed Rigid Format must reside as a separate file on the CDC and DEC VAX versions, as a member of a PDS on the IBM version and as a file or file.element on the UNIVAC version.

Before developing their own Rigid Formats, users are strongly advised to carefully study and examine the COSMIC-supplied Rigid Formats, particularly with regard to their use of the control cards. The following important guidelines should help users in developing their own Rigid Formats.

1. The DMAP sequence of the user Rigid Format must be tested for its correctness and logic. This testing may be done either in a DMAP environment or in the environment of an existing Rigid Format by use of ALTERs.

2. The card name table (the second part of a Rigid Format entry) must be constructed by assigning numbers 1 through 93 for all types of Case Control and Bulk Data Deck changes that will affect the logic of the Rigid Format. Normally, those input data changes that have the same effect on the logic of the Rigid Format are assigned the same number.

3. The file name table (the third part of a Rigid Format entry) must be constructed by assigning numbers 94 through 186 for all files (or data blocks) that are output by the functional modules in the Rigid Format. Normally, all files (or data blocks) output from a given functional module are assigned the same number.

4. The DMAP part (the first part of a Rigid Format entry) must be constructed by following each statement in the DMAP sequence by the appropriate control cards and by ensuring that each DMAP entry (that is, a DMAP statement and its associated control cards) is separated from the next DMAP entry by at least one comment card.

5. A given DMAP statement must be followed by a "SBST" control card if that DMAP statement belongs to one or

more of the DMAP subsets. These subset numbers must be specified on the "SBST" card. The acceptable subset numbers and their meanings are documented under the description of the SOL Executive Control card in the NASTRAN User's Manual (Reference 2).

6. A "RFMT" control card must follow a DMAP statement if that DMAP instruction is to be flagged for execution on restart from a checkpoint of one of the COSMIC-supplied Rigid Formats. (It is not possible to have a restart in a COSMIC-supplied Rigid Format from a checkpoint of an user-developed Rigid Format.) This will be so if this DMAP instruction is not part of the DMAP sequence of the Rigid Format that was used in the checkpoint run. The "RFMT" control card must list the numbers of the appropriate COSMIC-supplied Rigid Formats (187 through 201 for Rigid Formats 1 through 15 for APPROACH DISP, 207, 208 and 209 for Rigid Formats 1, 3 and 9 for APPROACH HEAT and 214 and 215 for Rigid Formats 10 and 11 for APPROACH AERO).

7. A DMAP statement must be followed by one or more "CARD" control cards indicating the effective input data changes that require that DMAP instruction to be flagged for execution on restart. Any effective input data change will affect one or more files (or data blocks) or parameters in the DMAP sequence. Therefore, for a given data change, all DMAP instructions that use the affected files (or data blocks) or parameters as input are potential candidates to be flagged for execution on restart. However, the logic of these individual DMAP instructions must be checked further (see Reference 7) to see if they are really impacted by the given data change. This procedure must be applied in turn to those DMAP instructions that use the output of the affected DMAP instructions as input. This procedure must be repeated until the entire DMAP sequence has been considered.

8. A DMAP statement must be followed by one or more "FILE" control cards indicating the DMAP files (or data blocks) whose generation requires the execution flag for that DMAP statement to be turned on during restart. Normally, for a given DMAP file (or data block) that is required on restart but is not available from the checkpoint run, the DMAP instruction that generated it must be flagged for execution. However, in practice, additional DMAP instructions like PURGE and EQUIV that manipulate the given file (or data block) must also be flagged for execution.

9. The restart flags for a COND DMAP instruction must include the restart flags for those DMAP instructions whose execution it controls.

10. "PHS1", "PHS2" and "PHS3" control cards must not be used as the substructure capability is not applicable to user Rigid Formats.

Usage of User-Developed Rigid Formats

An user-developed Rigid Format is referenced through the use of the "SOL" card in the Executive Control Deck. However, instead of specifying the solution number or the name of the COSMIC-supplied Rigid Format on this card, the name of the user-developed Rigid Format is specified. This name is a file name on the CDC and DEC VAX versions, a member name of a PDS on the IBM version and a file or file.element name on the UNIVAC version. The member name given on the IBM version must be in the file referenced on the RFDATA DD statement. The manner in which an user-developed Rigid Format is accessed and used is similar to that of a COSMIC-supplied Rigid Format, as explained in the examples given above. Thus, for instance, an user-developed Rigid Format can be accessed and used on the CDC version in the following manner.

```
/JOB.
.
.
GET,NEWRF.
RFL,220000.
REDUCE,-.
LINK1,INPUT,OUTPUT,PUNCH,UT1.
/EOR
ID ....
SOL NEWRF
.
.
/EOF
```

User Advantages

The advantages of the Rigid Format Data Base are readily apparent and are discussed in detail in Reference 1. Users can now very easily update the data base to incorporate corrections due to Software Problem Reports (SPRs) relating to Rigid Formats and their associated restart and subset tables. This ease also benefits the maintenance contractor in the maintenance of the Rigid Formats. Further, users can now generate their own Rigid Formats with restart capabilities or modify existing Rigid Formats permanently for their own use. Previously, changes to Rigid Formats required the use of temporary DMAP ALTERs or Fortran compilations and the link editing of NASTRAN. Elimination of these compilations and link edits benefits both the user and the maintenance contractor.

THE "READFILE" CAPABILITY

The "READFILE" capability allows a user to logically read data from one or more external, secondary, card-image files by referencing these files from the NASTRAN primary input file. The primary input file is the file that is assigned to Fortran unit 5 from which NASTRAN normally reads the input data.

Description of the Capability

The format of the "READFILE" card is as follows:

```
READFILE    name
```

where "name" refers to an external, secondary, card-image file.

When a "READFILE" card is encountered in the primary input file, NASTRAN reads all subsequent input data from the specified secondary file until an end-of-file condition or an ENDDATA card is encountered on that file, whichever occurs earlier. If an end-of-file condition is encountered on the secondary file before an ENDDATA card is detected, the program resumes reading of the input data from the primary input file and the process continues. If an ENDDATA card is encountered on the secondary file before an end-of-file condition is detected, obviously the program will not read any more input data from either the secondary file or the primary file, unless the INPUT module is being used in which case the data required for the INPUT module will be read from the primary input file (see Item 6 in the following discussion).

The following important points about the usage of the "READFILE" capability must be noted by the user:

1. The format of the "READFILE" card is free-field. The only restrictions are that there should be at least one space between the word "READFILE" and the "name" of the secondary file and that the card cannot extend beyond one card image (80 columns).

2. "READFILE" cards are permitted only in the NASTRAN primary input file and are not allowed in secondary input files. In other words, all references to secondary input files must be made from the primary input file and no secondary file can reference another secondary file.

3. As a corollary to the above, since a SOL card in the Executive Control Deck references an external, secondary file (either implicitly or explicitly), it must appear on the primary input file and is not permitted on a secondary file.

4. Any number of "READFILE" cards may be used, but each such card must reference an unique secondary file name.

5. "READFILE" cards may be used anywhere in the Executive Control, Substructure Control, Case Control and Bulk Data Decks. (The NASTRAN card can also be specified on a secondary file.)

6. If the INPUT module is used, the data required for that module must appear on the primary input file.

7. On the CDC and DEC VAX versions, "name" may be any valid file name (see Examples 1 and 2 below). On the IBM version, "name" may be either a sequential file name (see Example 3) or a member name of a PDS (see Example 4). On the UNIVAC, "name" may be any file name (see Example 5) or file.element name (see Example 6).

Examples of "READFILE" Capability Usage

The following examples illustrate several ways in which the "READFILE" capability can be used. These examples also illustrate the usage of this capability on all four versions of NASTRAN.

Example 1

This example illustrates the usage of the "READFILE" capability for reading in the restart dictionary in a checkpoint/restart run on the CDC version. (This example assumes that the output on the punch file in the checkpoint run contains only the restart dictionary.)

```
/JOB
.
.
COPYBR,INPUT,INPUT1.
COPYBR,INPUT,INPUT2.
REWIND,INPUT1,INPUT2.
* RUN CHECKPOINT JOB
LINK1,INPUT1,OUTPUT,PUNCH1,UT1.
* MANIPULATE FILES
PACK,PUNCH1.
REWIND,PUNCH1.
RETURN,POOL.
RENAME,OPTP=NPTP.
* RUN RESTART JOB
LINK1,INPUT2,OUTPUT,PUNCH2,UT1.
/EOR
NASTRAN  FILES=NPTP
.
. (DATA FOR CHECKPOINT JOB)
.
```

```
/EOR
NASTRAN  FILES=OPTP
.
$ READ THE RESTART DICTIONARY
READFILE PUNCH1
.
CEND
.
.   (DATA FOR RESTART JOB)
.
/EOF
```

Example 2

This example illustrates the use of multiple "READFILE" cards on the DEC VAX version.

```
ID ....
.
.
BEGIN BULK
READFILE DDB1:[NASDIR]FUSELAGE.DT
READFILE DDB1:[NASDIR]WINGS.DT
READFILE DDB1:[NASDIR]TAIL.DT
ENDDATA
```

The directory and device names need not be specified if default values are to be used.

Example 3

In this example, the "READFILE" capability is used to access a sequential file on the IBM version. The format for reading a sequential file is to include the DDname of the file on the "READFILE" card as shown below.

```
//  EXEC NASTRAN
//NS.CARDS  DD   DSN=USER.JOB1EXEC.DATA,DISP=SHR
//NS.SYSIN  DD   *
ID ....
.
.
READFILE    CARDS
/*
```

An ENDDATA card is not used in the Bulk Data Deck here as it is assumed to be included in the data on the sequential file.

Example 4

In this example, the "READFILE" capability is used to read a member of a PDS on the IBM version. The format for reading a member of a PDS is to include the DDname of the PDS with the member name in parentheses immediately following it as shown below.

```
//  EXEC NASTRAN
//NS.CARDS  DD   DSN=USER.PDS.DATA,DISP=SHR
//NS.SYSIN  DD   *
ID ....
.
.
READFILE    CARDS(JOB2EXEC)
.
.
/*
```

The member JOB2EXEC is read from the PDS USER.PDS.DATA.

Example 5

In this example, a file name on the UNIVAC is referenced by a "READFILE" card.

```
@ASG,A CARDS*UN1EXEC.
@XQT   *NASTRAN.LINK1
ID ....
READFILE   CARDS*UN1EXEC.
.
.
```

The file UN1EXEC with the qualifier CARDS will be read immediately after the ID card.

Example 6

In this example, a file.element name on the UNIVAC is referenced by a "READFILE" card.

```
@ASG,A CARDS*UN2.
@XQT   *NASTRAN.LINK1
ID ....
READFILE   CARDS*UN2.EXEC
.
.
```

The element EXEC of file UN2 with the qualifier CARDS is read immediately after the ID card.

CONCLUDING REMARKS

The Rigid Format Data Base and the "READFILE" capability will be welcomed by both the NASTRAN user community and the maintenance contractor. The Rigid Format Data Base is easily maintained and allows users the freedom of updating and modifying existing Rigid Formats as well as generating their own Rigid Formats, without having to compile and link edit NASTRAN. The "READFILE" capability will also prove to be extremely helpful and convenient to users. Both features will greatly enhance the flexibility, generality and attractiveness of NASTRAN.

REFERENCES

1. Pamidi, P. R., and Brown, K. B., "A New Method for Generating and Maintaining Rigid Formats in NASTRAN," NASA Conference Publication 2284, Eleventh NASTRAN Users' Colloquium, San Francisco, California, May 5-6, 1983, pp. 249-254.

2. The NASTRAN User's Manual, NASA SP-222(06), September 1983.

3. Control Data Corporation NOS 1.0 Reference Manual, Document No. 60435400.

4. VAX/VMS Command Language User's Guide, Digital Equipment Corporation, Order No. AA-D023C-TE.

5. IBM OS/VS2 MVS JCL, Document No. GC28-0692-4.

6. Sperry UNIVAC 1100 Series Executive System EXEC Programmer Reference, Volume 2, Document No. UP-4144.2.

7. The NASTRAN Programmer's Manual, NASA SP-223(05), December 1978.