Evaluation of the New Database in Version 66

Ingo Raasch
BMW AG, Munich, West Germany
(currently NASA Ames Research Center)

Introduction:

At BMW most MSC/NASTRAN runs are single shot runs, which save a minimum amount of data on the Data Recovery Database (1). Its size is about 3-5% of a restart tape or a normal superelement database. Postprocessing is initiated by an inhouse program, which reads this database and sets up a new Nastran job, which in turn creates all the necessary datablocks the user needs for further processing. Because the interfacing of a finite element program with a postprocessor is error prone, it was possible to reduce the number of unsuccessful runs considerably. This user friendly environment needs to be implemented in Version 66. Documentation of the new database does not exist, similarly to the previous versions, therefore the structure of the database was extracted from many test runs and dumps. Description of the the previous version of the database was met by considerable interest by the MSC/NASTRAN user community (2), therefore it is worthwhile to document our experience with the new database.

Database Structure:

The new database consists of at least two direct access files with a record size of BUFFSIZE. The first is the MASTER, which contains all the NDDL-relevant data, and the second is one or more DBALLi, which store the more traditional datablocks. The MASTER is needed at all times whereas one can be selective with the need for DBALLi.

Each physical record of any file consists of a partial or one or more logical records. Each logical record is surrounded and/or continued by hexadecimal keywords. However, these keywords are not always used consistently. The various keywords are described in detail in Appendix A. Usually each physical record is preceded by three header words:

- Physical record number in the datablock
- Words used in the physical record
- One word for an unidentified purpose

All records have a word structure, i.e. a word is a integer word, which can contain an integer, 4 characters or a real number. Two words contain a double precision number and four words a complex number. The first record of each database file contains information such as the creation/revision time stamp, logical name, file name, BUFFSIZE, and allocated, but not necessarily used, number of physical records. This record has no proper

keywords at the beginning but is properly closed. The encryption of date and time is different in the first record than in the remainder of the database.

The next physical record is always the beginning of the file's dictionary. It consists of 30 word 'logical' records, where two of these records make up an entry in the dictionary. These entries contain either information about a datablock, like the datablock trailer and where the datablock is stored, or the name, type and value of a NDDL parameter. In the latter case, the entry is 60 words long but only 11 words are used at most. Each logical record starts with a record name and a record pointer. They are made up of the physical record number within the datablock and the word where the logical record starts in the physical record. The first record name of a datablock entry is used later as a dataset pointer. The first logical record of this datablock is always empty, but its record pointer points to the 'next' empty record (this does not mean it is physically the next empty record). All empty records are concatenated. The datablocks can span over more than one physical record and the dictionary entry will point to the location and the number of records at that location. However, the dictionary is an open ended datablock, i.e. it does not have proper keywords at the end. Things get complicated even more; not all of the physical records indicated in the dictionary entry are written. Therefore, the way to find the 'end' of this datablock is to look for an empty record, which does not concatenate to any other record. The default size of this datablock is 10 records no matter how large the BUFFSIZE is. The third physical record is always a map of the used physical records.

The MASTER database contains the NDDL description, which is written to the database during the initialization of the database and freezes the NDDL description. NDDL defines where each datablock and each NDDL parameter resides. For each datablock there is a path with current values of its qualifiers. However, this information is not stored together with the datablock, but is available only through a complicated system of pointers, which so far could not be resolved rationally. The dataset pointer points to an entry in the DBENTRY datablock in the MASTER, which indicates the dataset key. The dataset key is a unique number for a given path and its qualifier values. The resolution of the key into the proper path qualifiers and its values is hidden in the datablock PATHQUAL. The resolution found so far, which works in all the test cases, lacks rationality and, therefore, will not described here.

Writing DMAP in Version 66:

In this section, points which relate to the use of the database in a DMAP program will be considered. Previously, a datablock was used to transfer data from one DMAP module to another. Storing and retrieving of datablocks was done through a module. In Version 66 there is the distinction between a local datablock, which resides on temporary scratch disks, and database datablocks. Only the latter can be, but need not be, saved on a database. However, these datablocks need a cumbersome, not always transparent, definition in the NDDL **and** in the DMAP, i.e.:

- In the NDDL there is a definition of the
    - datablock name
    - path, a collection of qualifiers.
- In the DMAP the datablock must be TYPEd

Storing and retrieval is done automatically. The datablock is identified by its name, path, current values of qualifiers, current version and project. If any of these many items do not agree, one might recover by using the DBVIEW command.

NDDL, old NDDL, and user's NDDL:

The first thing to remember with any of the different types of NDDL is: each NDDL creates a unique incompatible database system, which cannot be used by any other NDDL version. MSC supplies two NDDL versions. The old NDDL, which works with the old solution sequences. The non-superelement sequences do not use the database at all and there are no automatic restart options. Unfortunately, MSC announced that it will discontinue the support of old NDDL. MSC favors the NDDL, which was developed for the new solution sequences. This NDDL includes more datablock names (remember each datablock name which should be stored must be defined in the NDDL!). It also includes a datablock description of many tables, which are important for automatic restarts. However, not all of the descriptions are up to date or self-contained. Furthermore, the NDDL contains the dependency of the datablocks for automatic restarts. Anyone who is used to doing extensive DMAPing is faced with the prospect of creating his own NDDL when the database is used as a storage device. This implies that one must create his own incompatible NASTRAN world or universe with many incompatible planets.

Database Size:

The new NDDL, which is stored with each database, has to be the same for all compatible runs. This creates a considerable overhead, which for small problems, is more than the actual data of the finite element problem, in particular since some of the default values are generated very generously. Similarly, it is difficult to keep the database clean, e.g. for SOL200 runs it was observed that in the dictionaries there are about 50% more entries than DBDIR will show, i.e.useless leftovers. One improvement in the new database design is that the space on the database for a given datablock no longer has to be contiguous, which will be appreciated by many users. Each use of an existing database will add a new version and will not overwrite any existing datablocks. Database housekeeping has to be performed by the user, which seems to be unreliable and unfriendly method.

Restarts:

Much development effort was expanded in generating automatic restart capabilities, even though many users do not think highly of automatic restarts. Some points about the value of automatic restart capabilities should be stated:

- Computer power continues to increase everywhere
- Disc space is never in abundance and will remain restricted
- Likelihood of an error increases with restarts
- Likelihood of resolving errors decreases with restarts
- Certainty of problem solution decreases with restarts
- Wall clock time is more important than cpu time
- There are two types of restarts:
    - continuation of the analysis
    - change of input data.

In the latter case, author's experience is that more than 90% of restarts virtually do all the calculation again and do not use any of the stored data except the BULK DATA DECK. In the former case there is a lot of data stored which never will used again. Single shot run which saves no data needs approximately 1/4 of database needed for a superelement database without superelements. This should also be a reasonable size for restart databases.

Quality Assurance:

Quality of a program starts with quality of design and programming. User friendliness and quality control are closely connected. Results which are hard to read are very difficult to analyze for their correctness. In this respect, some of the features of the new database are designed in a way which might be complete but hardly meets the requirements of userfriendliness, e.g. the output of the DBDIR, an example is shown in Appendix B. Since a datablock is identified by its name and the qualifier values of its path, one is very soon flipping pages back and forth in order to identify a datablock. An output like the one shown in Appendix C will better serve this purpose. DIAG 8 still indicates when a matrix is generated but its path and its qualifier values are not available. In the database there may be many datablocks with the same name and path but different qualifier values. However, there is no way to get a condensed listing of just these datablocks along with their qualifiers. One can get all or nothing.

Consistency in the code and maintainability are interdependent and will affect the stability of the program in the long run. Therefore, inconsistencies in the coding might be one indicator for the quality of a product.

Finite element analysis in the future will be considered to be more a tool than an art. The quality of an analysis will have to assured, and when it is questioned, it will have to be easily rectified. The operations that are carried out should be transparent and understandable to the user, who is a craftsman and not an artist. In view of these concerns, Version 66 is not an improvement at the present time.

Conclusions:

- MSC/NASTRAN Version 66 is new program which is input compatible to MSC/NASTRAN Version 65 or previous versions.
- MSC/NASTRAN Version 66 is not one program but at least two incompatible programs which are, again, only input compatible.
- The database is still a direct access file. The smallest addressable item is a datablock (with the exception of some NDDL datablocks).
- Some of the logical record keys contain addresses, which are BUFFSIZE dependent. Therefore, there is no way to combine databases of different BUFFSIZEs in the future.
- The design of the database is not rigorous and self-contained. There are datablocks which violate datablock syntax, such as the first physical record, DBENTRY (first record and end of file), DBDIR (end of file), or NDDL (description of CASECC and some of the data items are presented in reverse order). This will make the program harder to maintain.
- The flexibility of DMAPing is strongly curtailed by the inflexibility of the NDDL, as it determines the compatibility.
- The database operations are no longer transparent to the user and debugging tool like old DIAG 20 are not provided.
- Many details of the new database are not very userfriendly and they are hard to utilize. It is the hope for the future that MSC will request for user feedback on the proposed designs before casting them into the final form.
- Version 66 is a first step into the brave new world of NASTRAN. Yet, neither the user nor MSC reached the new frontier.

Acknowledgement:

The author would like to thank Mr. Mladen Chargin of Ames Research Center for his encouragement and editorial efforts to get this paper done.

References:

1. Schulze Schwering, W.; Raasch, I., "Efficient Data Recovery and Postprocessing with Nastran," Nastran User's Conference, 1981, Munich, West Germany.

2. Raasch, I.; Sielaff, J., "The User Interface for MSC/NASTRAN on an Inhomogeneous Hardware Environment," Nastran User's Conference, 1989, London, Great Britain.

Appendix A :

Data Block Description:

The database consists of at least two files: MASTER and DBALL. Both have essentially the same structure. They are direct access files with a record length of BUFFSIZE. The first and the second record have to be accessed directly and they always contain the same information. The first record contains some general information whereas the second record contains the start of the database dictionary, which describes the contents of records 2 to n.

The Master contains all the NDDL information and all the entries are tables. The DBALL contains all the DMAP related information, i.e. data blocks and parameters declared in the NDDL.

Each physical record has three header words:

- physical record number in the data block
- words used in the physical record
- unidentified purpose

The physical record is broken up into logical records. This is done by use of certain seperator keys. Each key consists of a keytype and two integer arguments:

- 8 left most bits : keytype
- following 8 bits : key argument 2
- following 16 bits : key argument 1

Key Type hex	Key Arg 1 integer	Key Arg 2 integer	Description
11	length		begining of table record
22			begining of matrix record
31	start address		end of complete record
32	start address	record no.	end of spanned record
33	start address	record no.	intermediate end of spanned record
55			intermediate end of spanned file
77			end of file
44			start of matrix column
88			start of matrix string
66			end of matrix column

1. Record of Database:

Word	Type	Description
1		
2		
3		
4		
5	integer	BUFFSIZE
•		
•		
17	integer	creation date
18	interger	creation time
19	integer	revision date
20	integer	revision time
21		
22		
23		
24	character	file name
25		
•		
•		
•		
34-65	character	dataset name
66	integer	allocated number of blocks
rest		unidentified

It ends with EOR-key and EOF-key. Note that the date and time is encrypted differently than in any of the following data blocks (**great**).

DBDIR: Database Dictionary

This data block always starts with the second physical record of the data base and it does not have a header record as normal data blocks have. The first logical record is always a empty record and its record pointer concatenates to the "next" empty record.

First Record (Data Block Entry)		
Word	Type	Description
1	integer	record name (16 bit record number, 16 bit word number) dataset pointer
2	integer	record pointer (points to continuation record)
3-4	character	data block name
5-15	integer	matrix trailer
•		
24	integer	number of physical records in data block
Second Record (Data Block Entry)		
1	integer	record name (16 bit record number, 16 bit word number)
2	integer	record pointer (points to continuation record)
5	integer	address of first physical record of data block
		if < 0 only one record if > 0 next word: number of records. These entries are repeated as many times as necessary.If this entry is a NDDL parameter also two records are written butmost of it is empty.

First Record (NDDL Parameter Entry)		
Word No.	Type	Description
1	integer	record name (16 bit record number, 16 bit word number)
2	integer	record pointer (points to continuation record)
3	integer	0
4-5	character	parameter name
6	integer	type
7	integer	length
8+length-1	type	value
rest		wasted

The physical records all end with 55-key and do not have a EOF-key. The second record is present but empty (**great**).

DBNAME: Data Block Names

1. record		
Word	Type	Description
1-2	character	data block name
3	integer	path
4	integer	unidentified
5	integer	location
6	integer	address in DATABLOC upper 16 bit record number lower 16 bit word number
These 6 word are repeated as many times as there are data blocks in the NDDL.		

PROJVERS: Project and Version

1. Record

Word	Type	Description
1	integer	number of projects
2	integer	project number
3-12	character	project title
Words 2-12 are repeated as many times as there are projects.		

2. Record

Word	Type	Description
1	integer	number of versions
2	integer	project number
3	integer	version number
4	integer	creation date
5	integer	creation time
words 2-5 are repeated as many times as there are versions.		

DBENTRY: Data File Pointers

This data block does not have a header record as normal data blocks have. The first record is always a empty record. However its record pointer is not used to point to the "next" empty record but instead word number 5 (**great**).

Record description

Word	Type	Description
1	integer	record name
2	integer	data block status 0 unique 1 primary 2 equivalenced 4 purged
3	integer	unidentified
4	integer	creation date
5	integer	creation time
6	integer	revision date
7	integer	revision time
8	integer	version
9	integer	
10	integer	project number
11	integer	
12-13	character	blank
14	integer	record pointer to next record used for empty records or address of parent DB entry
15	integer	address for entry for same DB but different path
16	integer	GINO address
17	integer	dataset pointer or address in DBDIR
18	integer	dataset key
This data block has 55-keys but does **not** end with a EOF-key (**great**).		

PATHQUAL: Path Qualifiers

1. record qualifiers

Word	Type	Description
1-2	character	qualifier name
3	integer	type
4	integer	length
5	integer	address
These 5 words are repeated as many times as there are qualifiers.		

2. Record data block names (almost the same as data block DBNAME)

Word	Type	Description
1-2	character	data block name
3	integer	path
4	integer	address in DBENTRY
5	integer	location
6	integer	address in DATABLOC
These 6 words are repeated as many times as there are data blocks.		

3. Record paths

Word	Type	Description
1		
15		
16	integer	number of qualifiers per key
17	integer	address of qualifier
18	integer	
19		

4. Record NDDL parameter definition

Word	Type	Description
1-2	character	NDDL parameter name
•		
•		
9		
These 9 words are repeated as many times as there are NDDL parameters.		

5. Record default values of qualifiers

6. Record default values of NDDL parameter

7. Record current qualifier values

Word	Type	Description
1		
11		unknown
12	integer	address of qualifier values for key = j
j	integer	key
j+1	integer	address of key with same qualifiers = k
j+2-j+m+1	variable	qualifier values

DATABLOC: Data Block Description

Word	Type	Description
1-2	character	NDDL name
3	integer	data block type
		1table with fixed number of records 3 matrix 5 table with variable number of records 16 cases table
4	integer	number of record descriptors = j
5-6	character	record name
7-9	integer	record header words
10	integer	address of data type description +4 (**great**) = k
These 6 words are repeated j times.		
k	integer	number of words in the data type description
k+1	integer	info type
if info type =1		
k+2	integer	data type 1 = I; 2 = RS; 3 = Char4
k+3	integer	data item
k+4	integer	0
if info type =3		
k+2	integer	data type
k+3	integer	data item
k+4	integer	number of words
if info type =4 (either)		
k+2	integer	number of words including the previous in the either description
k+3	integer	either label
k+4	integer	0
if info type =5 (or)		
k+2	integer	number of words including the previous in the or description
k+3	integer	or label
k+4	integer	0
if info type =6 (begin of entry loop)		
k+2	integer	number of words in the entry loop not including this word (**great**)

if info type =7 (begin of label entry loop)		
k+2	integer	
k+3,+4	integer	entry name
k+5	integer	number of words in the entry loop not including this word.
if info type =8 (end of entry loop)		
k+2	integer	number of words of EOR description
k+...	integer	EOR mark
if info type =9 (end of entry with count)		
k+2	integer	count for entry
if info type =10 (unused words)		
k+2	integer	number of unused words
•		
l	integer	number of words for the data item descriptor
l+1,+2	integer	data item descriptor in **reverse order !!!!**

There is additional information for same tables of the form

3, 11, record type

however there is no path to this information.

Appendix B

NDDL DATABLOCK LISTING

ENTRY	NAME	DBSET	PROJECT ID ^	VERSION ID ^	CREATION DATE	REVISION DATE	NO. BLOCKS	KEY NUMBER	EQUIV/ PURGE	FILE POINTER
1	AXIC	DBALL	1	2	1/ 8/90 10:11. 7	1/ 8/90 10:11. 7	0	213	0/1	66468
2	BAA	DBALL	1	2	1/ 8/90 10:11.32	1/ 8/90 10:11.32	0	303	0/1	397572
3	BAA	DBALL	1	2	1/ 8/90 10:11.12	1/ 8/90 10:11.12	0	246	0/1	200100
4	BAA	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	0	265	0/1	266052
5	BAA	DBALL	1	2	1/ 8/90 10:11.46	1/ 8/90 10:11.46	0	322	0/1	524740
6	BAA	DBALL	1	2	1/ 8/90 10:11.22	1/ 8/90 10:11.22	0	284	0/1	332004
7	BGG	DBALL	1	2	1/ 8/90 10:11.46	1/ 8/90 10:11.46	1	307	0/1	524676
8	BGG	DBALL	1	2	1/ 8/90 10:11.32	1/ 8/90 10:11.32	0	288	0/1	397508
9	BGG	DBALL	1	2	1/ 8/90 10:11.22	1/ 8/90 10:11.22	0	269	0/1	331940
10	BGG	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	0	250	0/1	265988
11	BGG	DBALL	1	2	1/ 8/90 10:11.12	1/ 8/90 10:11.12	0	231	0/1	200036
12	BGPDTS	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	220	2/0	68836
13	BGPDTS	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	222	2/0	70052
14	BGPDTS	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	224	2/0	132228
15	BGPDTS	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	1	226	2/0	133444
16	BGPDTS	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	1	228	0/0	134212
17	BGPDTX	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	1	228	0/0	135236
18	BGPDTX	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	224	1/0	132228
19	BGPDTX	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	222	1/0	70052
20	BGPDTX	DBALL	1	2	1/ 8/90 10:11. 9	1/ 8/90 10:11. 9	1	220	1/0	68836
21	BGPDTX	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	1	226	1/0	133444
22	BJJ	DBALL	1	2	1/ 8/90 10:11.27	1/ 8/90 10:11.27	0	288	0/1	395204
23	BJJ	DBALL	1	2	1/ 8/90 10:11.41	1/ 8/90 10:11.41	0	307	0/1	460772
24	BJJ	DBALL	1	2	1/ 8/90 10:11.11	1/ 8/90 10:11.11	0	231	0/1	197348
25	BJJ	DBALL	1	2	1/ 8/90 10:11.13	1/ 8/90 10:11.13	0	250	0/1	263300
26	BJJ	DBALL	1	2	1/ 8/90 10:11.17	1/ 8/90 10:11.17	0	269	0/1	329252
27	BULK	DBALL	1	2	1/ 8/90 10:11. 6	1/ 8/90 10:11. 6	1	213	0/0	65764
28	CASECC	DBALL	1	2	1/ 8/90 10:11. 8	1/ 8/90 10:11. 8	1	215	0/0	67300
29	CASECC	DBALL	1	2	1/ 8/90 10:11. 8	1/ 8/90 10:11. 8	1	216	0/0	67236

NDDL DATABLOCK TRAILER RECORD LISTING

ENTRY	DATBLOCK NAME	TRAILER									
1	AXIC	0	0	0	0	0	0	0	0	0	0
2	BAA	0	0	0	0	0	0	0	0	0	0
3	BAA	0	0	0	0	0	0	0	0	0	0
4	BAA	0	0	0	0	0	0	0	0	0	0
5	BAA	0	0	0	0	0	0	0	0	0	0
6	BAA	0	0	0	0	0	0	0	0	0	0
7	BGG	0	0	0	0	0	0	0	0	0	0
8	BGG	0	0	0	0	0	0	0	0	0	0
9	BGG	0	0	0	0	0	0	0	0	0	0
10	BGG	0	0	0	0	0	0	0	0	0	0
11	BGG	0	0	0	0	0	0	0	0	0	0
12	BGPDTS	205	6	0	0	0	0	0	0	0	0
13	BGPDTS	205	6	0	0	0	0	0	0	0	0
14	BGPDTS	205	6	0	0	0	0	0	0	0	0
15	BGPDTS	205	2	0	0	0	0	0	0	0	0
16	BGPDTS	201	35	0	0	0	0	0	1	1	0
17	BGPDTX	205	39	0	0	0	0	0	0	0	0
18	BGPDTX	205	6	0	0	0	0	0	0	0	0
19	BGPDTX	205	6	0	0	0	0	0	0	0	0
20	BGPDTX	205	6	0	0	0	0	0	0	0	0
21	BGPDTX	205	2	0	0	0	0	0	0	0	0
22	BJJ	0	0	0	0	0	0	0	0	0	0
23	BJJ	0	0	0	0	0	0	0	0	0	0
24	BJJ	0	0	0	0	0	0	0	0	0	0
25	BJJ	0	0	0	0	0	0	0	0	0	0
26	BJJ	0	0	0	0	0	0	0	0	0	0
27	BULK	201	1	0	0	0	0	0	0	0	0
28	CASECC	101	2	2	338	0	0	0	0	0	0
29	CASECC	101	3	3	338	0	0	0	0	0	0

NDDL PARAMETER LISTING

ENTRY	NAME	DBSET	PROJECT ID ^	VERSION ID ^	CREATION DATE	REVISION DATE	KEY NUMBER
1	BCHNG	DBALL	1	2	1/ 8/90 10:11.12	1/ 8/90 10:11.13	220
2	BCHNG	DBALL	1	2	1/ 8/90 10:11.46	1/ 8/90 10:11.47	228
3	BCHNG	DBALL	1	2	1/ 8/90 10:11.32	1/ 8/90 10:11.34	226
4	BCHNG	DBALL	1	2	1/ 8/90 10:11.22	1/ 8/90 10:11.23	224
5	BCHNG	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	222
6	EPSBIG	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	219
7	EPSBIG	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	223
8	EPSBIG	DBALL	1	2	1/ 8/90 10:11.23	1/ 8/90 10:11.23	225
9	EPSBIG	DBALL	1	2	1/ 8/90 10:11.34	1/ 8/90 10:11.34	227
10	EPSBIG	DBALL	1	2	1/ 8/90 10:11.13	1/ 8/90 10:11.13	221
11	ERROR	DBALL	1	2	1/ 8/90 10:11.10	1/ 8/90 10:11.10	214
12	GOODVER	DBALL	1	2	1/ 8/90 10:11. 6	1/ 8/90 10:11.10	213
13	HNNLK	DBALL	1	2	1/ 8/90 10:11.38	1/ 8/90 10:11.38	305
14	HNNLK	DBALL	1	2	1/ 8/90 10:11.11	1/ 8/90 10:11.11	229
15	HNNLK	DBALL	1	2	1/ 8/90 10:11.13	1/ 8/90 10:11.13	248
16	HNNLK	DBALL	1	2	1/ 8/90 10:11.15	1/ 8/90 10:11.15	267
17	HNNLK	DBALL	1	2	1/ 8/90 10:11.27	1/ 8/90 10:11.27	286
18	INRLM	DBALL	1	2	1/ 8/90 10:11.13	1/ 8/90 10:11.13	221
19	INRLM	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	223
20	INRLM	DBALL	1	2	1/ 8/90 10:11.24	1/ 8/90 10:11.24	225
21	INRLM	DBALL	1	2	1/ 8/90 10:11.34	1/ 8/90 10:11.34	227
22	INRLM	DBALL	1	2	1/ 8/90 10:11.11	1/ 8/90 10:11.11	219
23	K4CHNG	DBALL	1	2	1/ 8/90 10:11.12	1/ 8/90 10:11.13	220
24	K4CHNG	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	222
25	K4CHNG	DBALL	1	2	1/ 8/90 10:11.22	1/ 8/90 10:11.23	224
26	K4CHNG	DBALL	1	2	1/ 8/90 10:11.32	1/ 8/90 10:11.33	226
27	K4CHNG	DBALL	1	2	1/ 8/90 10:11.47	1/ 8/90 10:11.47	228
28	KCHNG	DBALL	1	2	1/ 8/90 10:11.31	1/ 8/90 10:11.33	226
29	KCHNG	DBALL	1	2	1/ 8/90 10:11.12	1/ 8/90 10:11.13	220
30	KCHNG	DBALL	1	2	1/ 8/90 10:11.14	1/ 8/90 10:11.14	222
31	KCHNG	DBALL	1	2	1/ 8/90 10:11.20	1/ 8/90 10:11.23	224
32	KCHNG	DBALL	1	2	1/ 8/90 10:11.44	1/ 8/90 10:11.44	228

PATH VALUE TABLE (KEY NUMBER CORRESPONDS TO KEY NUMBER ENTRY IN ABOVE NDDL DESCRIPTIONS)

```
  KEY NUMBER =   213        QUALIFIER NAME AND VALUE
                            HIGHQUAL=           0

  KEY NUMBER =   214        QUALIFIER NAME AND VALUE
                            APRCH   =
                            HIGHQUAL=           0

  KEY NUMBER =   215        QUALIFIER NAME AND VALUE
                            APPC    =
                            HIGHQUAL=           0

  KEY NUMBER =   216        QUALIFIER NAME AND VALUE
                            APPC    =STATICS
                            HIGHQUAL=           0

  KEY NUMBER =   217        QUALIFIER NAME AND VALUE
                            APPC    =MODES
                            HIGHQUAL=           0

  KEY NUMBER =   218        QUALIFIER NAME AND VALUE
                            APPC    =SHAPE
                            HIGHQUAL=           0

  KEY NUMBER =   219        QUALIFIER NAME AND VALUE
                            SEID    =          90
                            PEID    =          90
                            APRCH   =
                            HIGHQUAL=           0

  KEY NUMBER =   220        QUALIFIER NAME AND VALUE
```

Appendix C

P R O J E C T V E R S I O N D I R E C T O R Y

PROJECT ID	VERSION ID	CREATION TIME
* B L A N K *	1 1	12/ 1/89 16:25: 6

DATABLOCK LISTING

ENTRY	NAME	DBSET	PROJECT ID	VERSION ID	REVISION DATE	KEY	NO. BLKS	FIRST BLK	PATHQUALIFIERS
1	AXIC	DBALL	1	1	12/ 1/89 16:25:13	213	0	0	HIGHQUAL= 0
2	BAA	DBALL	1	1	12/ 1/89 16:25:59	237	0	0	SEID = 0 PEID = 0 MPC = 0 SPC = 1 METH = 0 DYRD = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 K2GG = 'blank' B2GG = 'blank'
3	BGG	DBALL	1	1	12/ 1/89 16:25:59	222	0	0	PEID = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 B2GG = 'blank'
4	BGPDTS	DBALL	1	1	12/ 1/89 16:25:59	219	1	56	PEID = 0 APRCH = 'blank' HIGHQUAL= 0
5	BGPDTS	DBALL	1	1	12/ 1/89 16:25:15	219	1	49	PEID = 0 APRCH = 'blank' HIGHQUAL= 0
6	BJJ	DBALL	1	1	12/ 1/89 16:25:59	222	0	0	PEID = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 B2GG = 'blank'
7	BLAMA	DBALL	1	1	12/ 1/89 16:26: 0	242	1	96	SEID = 0 PEID = 0 LOAD = 1 MPC = 0 SPC = 1 BMETH = 2 TEMPLD = 0 DEFORM = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 K2GG = 'blank' P2G = 'blank'
8	BPHA	DBALL	1	1	12/ 1/89 16:26: 0	242	1	97	SEID = 0 PEID = 0 LOAD = 1 MPC = 0 SPC = 1 BMETH = 2 TEMPLD = 0 DEFORM = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 K2GG = 'blank' P2G = 'blank'
9	BULK	DBALL	1	1	12/ 1/89 16:25:13	213	1	13	HIGHQUAL= 0
10	CASECC	DBALL	1	1	12/ 1/89 16:25:14	217	1	38	APPC =MODES HIGHQUAL= 0
11	CASECC	DBALL	1	1	12/ 1/89 16:25:14	216	1	36	APPC =BUCK HIGHQUAL= 0
12	CASECC	DBALL	1	1	12/ 1/89 16:25:14	215	1	34	APPC =STATICS HIGHQUAL= 0
13	CASES	DBALL	1	1	12/ 1/89 16:25:15	218	1	43	SEID = 0 PEID = 0 APRCH = 'blank' HIGHQUAL= 0
14	CMLAMA	DBALL	1	1	12/ 1/89 16:25:59	232	0	0	PEID = 0 MPC = 0 SPC = 1 METH = 0 DYRD = 0 MTEMP = 0 APRCH = 'blank' HIGHQUAL= 0 K2GG = 'blank' M2GG = 'blank'
15	CMPHA	DBALL	1	1	12/ 1/89 16:25:59	232	0	0	PEID = 0 MPC = 0 SPC = 1 METH = 0 DYRD = 0 MTEMP = 0

```
 NDDL PARAMETER LISTING

ENTRY NAME      VALUE          DBSET PROJECT VERSION  REVISION          KEY
                                      ID      ID       DATE                    PATHQUALIFIERS
   1  DBDN     =DBALL         MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   2  DBRCV    =DBALL         MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   3  DBUP     =DBALL         MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   4  GOADB    =SCRATCH       MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   5  IFP      =DBALL         MASTER      1     1  12/ 1/89 16:25:14  214 APRCH   = 'blank' HIGHQUAL=       0
   6  OTDN     =SCRATCH       MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   7  OTIFP    =SCRATCH       MASTER      1     1  12/ 1/89 16:25:14  214 APRCH   = 'blank' HIGHQUAL=       0
   8  OTUP     =SCRATCH       MASTER      1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   9  UNDPLTDB=SCRATCH        MASTER      1     1  12/ 1/89 16:25:14  214 APRCH   = 'blank' HIGHQUAL=       0


 NDDL PARAMETER LISTING

ENTRY NAME      VALUE          DBSET PROJECT VERSION  REVISION          KEY
                                      ID      ID       DATE                    PATHQUALIFIERS
   1  BCHNG    =       F      DBALL       1     1  12/ 1/89 16:26: 0  219 PEID    =       0 APRCH   = 'blank' HIGHQUAL=       0
   2  EPSBIG   = 0.100000E+13 DBALL       1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   3  ERROR    =      -1      DBALL       1     1  12/ 1/89 16:25:15  214 APRCH   = 'blank' HIGHQUAL=       0
   4  GOODVER  =       T      DBALL       1     1  12/ 1/89 16:25:15  213 HIGHQUAL=       0
   5  HNNLK    =       0      DBALL       1     1  12/ 1/89 16:25:59  220 PEID    =       0 MTEMP   =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   6  INRLM    =       1      DBALL       1     1  12/ 1/89 16:25:59  218 SEID    =       0 PEID    =       0 APRCH   = 'blank'
                                                                             HIGHQUAL=       0
   7  K4CHNG   =       F      DBALL       1     1  12/ 1/89 16:26: 0  219 PEID    =       0 APRCH   = 'blank' HIGHQUAL=       0
   8  KCHNG    =       F      DBALL       1     1  12/ 1/89 16:26: 0  219 PEID    =       0 APRCH   = 'blank' HIGHQUAL=       0
   9  LUSETS   =      24      DBALL       1     1  12/ 1/89 16:25:15  219 PEID    =       0 APRCH   = 'blank' HIGHQUAL=       0
```