

*CASE

The *CASE command provides a way of running multiple LS-DYNA analyses (or cases) sequentially by submitting a single input file. When *CASE commands are used to define multiple cases, some portions of the input will be shared by some or all of the cases and other portions will be unique to each case. Because the cases are run sequentially, the results from one case, e.g., a dynain file, can be used in the analysis of a different, subsequent case. Each case creates a unique set of output file names by prepending "casen." to the default file name, e.g., case101.d3plot, case102.glstat.

When the *CASE keyword appears in an input deck, it becomes necessary to append the word "CASE" to the LS-DYNA execution line. For example, an SMP LS-DYNA execution line might look something like

```
path_to_ls-dyna i=input.k ncpu=-4 CASE
```

An MPP LS-DYNA execution line might look something like

```
mpirun -np 4 path_to_mpp971 i=input.k CASE
```

The default behavior is to run all the cases defined in the input deck sequentially in increasing numeric order. If any of the cases fail to run to normal termination, the subsequent cases will not be run.

To run a subset of the cases defined in the input deck, specify the case ID numbers following the word "CASE" on the execution line, for example, "CASE = 10,20,47" will only run case IDs 10, 20 and 47, and skip any other cases. There can be NO spaces before or after "=" or ",". The specified cases will be run in increasing numeric order and not necessarily in the order given on the command line.

***CASE_{OPTION}**

Available options include:

<BLANK>

BEGIN_N

END_N

Purpose: Define a series of cases and perhaps subcases. The options *CASE_BEGIN_n and *CASE_END_n appear in pairs and n is a numeric ID of a subcase. Subcase IDs may be referenced by the *CASE command in defining a case. In other words, a case may consist of one or more subcases. All keywords appearing between *CASE_BEGIN_n and *CASE_END_n comprise subcase n. If no *CASE command is defined, then subcases defined by *CASE_BEGIN_n and *CASE_END_n then become cases. *CASE_BE-

*CASE

GIN/*CASE_END can be nested, overlapped, and disjointed. Examples below demonstrate the use of these options.

An alternative way of defining subcases is by appending the string “CID = *n*” to the end of any keyword command. Any keyword so tagged will then be active only for those cases that reference subcase *n*. There can be more than one space between the keyword and “CID = *n*”.

Any keyword in the input deck not associated with a subcase is active for all cases.

The following input syntax applies only to the *CASE command, not to *CASE_BEGIN/*CASE_END.

Card 1	1	2	3	4	5	6	7	8
Variable	CASEID		JOBID					
Type	I		C					
Default	none		none					

Command Line Argument Cards. Command line cards set additional command line arguments for the case CASEID (see Card 1, above). Include as many as needed, or as few as none. Command line cards end when the first character of the next card is numeric.

Card 2	1	2	3	4	5	6	7	8
Variable			COMMANDS					
Type			A					
Default			Not Required					

Subcase ID Cards. Define active subcase IDs for case CASEID (see Card 1, above). These cards continue until the next keyword ("**") card.

Card 3	1	2	3	4	5	6	7	8
Variable	SCID1	SCID2
Type	I	I	I	I	I	I	I	I
Default	none	none	none	none	none	none	none	none

VARIABLE	DESCRIPTION
CASEID	Identification number for case.
JOBID	Optional string (no spaces) to be used as the jobid for this case. If no JOBID is specified, the string CASEXX is used, where XX is the CASEID in field 1
COMMANDS	Command line arguments.
SCIDn	Subcase ID active for case CASEID.

Remarks:

1. If no *CASE keyword appears, subcases defined with *CASE_BEGIN/*CASE-END commands become cases and *CASE_BEGIN can optionally be followed by extra command line arguments.
2. If no *CASE keyword appears, it is an error to append "CID = *n*" to any keyword.
3. If multiple *CASE or *CASE_BEGIN keywords appear that have the same ID, their command line arguments and active commands are merged.
4. The *CASE or *CASE_BEGIN keywords cannot be used within an include (*INCLUDE) file.

Example 1:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$ Define case 101 which includes subcase 1.  
$ Define case 102 which includes subcase 4.
```

*CASE

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
*CASE  
$  
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7  
$    CASEID  
      101 JOBID_FOR_CASE101  
MEMORY=20M  
1  
$  
*CASE  
$    CASEID  
      102  
MEMORY=20M  NCYCLE=1845  
4  
$  
*TITLE CID=1  
THIS IS THE TITLE FOR CASE 101  
$  
*TITLE CID=4  
THIS IS THE TITLE FOR CASE 102  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example 2:

```
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
$$$$ Illustrate overlapping subcases.  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
$  
*CASE_BEGIN_5  
$...>....1....>....2....>....3....>....4....>....5....>....6....>....7....>  
*DATABASE_BINARY_D3THDT  
1.e-5  
*CASE_BEGIN_3  
*DATABASE_NODOUT  
1.e-5  
*CASE_END_5  
*DATABASE_ELOUT  
1.e-5  
*CASE_END_3  
$  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Example 2 above will generate d3thdt and nodout for CID = 5, and nodout and elout for CID = 3.