Unified Model Documentation Paper X2

The Nupdate Source Code Manager

By

Ramesh Krishna

Version number 2 dated 9th November 1993

For version 3.2 of the Unified Model

Numerical Weather Prediction
Meteorological Office
London Road
BRACKNELL
Berkshire
RG12 2SZ
United Kingdom

Modification Record		
Document Version	**Author**	**Description**
2	Tracey Smith	Changed references from update to nupdate where appropriate

Contents

1. Command Line Arguments

NAME

nupdate - Invokes portable NUPDATE utility

SYNOPSIS

nupdate -p pdf -i idf -c cdf [-a suffix] [-d ifp] [-f | -q decks] [-o opts]

The nupdate utility creates and modifies program libraries (PLs) and produces output that you can use as input to other programs, particularly compiler assemblers.

Options:

-p pdf PL (program library) file name. This file is the input Program Library. You must include the file name; -p alone is invalid.

-i idf Input file name (this is the mods file).

-c cdf Compile file name. The file name is appended with a period and a single letter as defined by default (.f, .s, .p, and .c) or specified by the -a option. The file name and the following period and suffix should not exceed 14 characters (thus, the input file name must not exceed 12 characters). The file must not exist, because it is created by nupdate. The default suffix is .f.

-a suffix This is the suffix that will be appended to the compile file name provided by the -c argument.

-d ifp Defines names (up to 8 characters) to be used with an IF directive. If you specify more than one defined name, use space as the separator character. Defined names need not be different from deck names, common deck names, or modification identifiers. They are known only in the nupdate run being processed. You can define up to 64 names with the -d option.

-f

-q decks Full, quick, or normal update run. You can specify either -f or -q option. This determines the compile file containing the source file, and the new PL contents.

-f Full update mode; all active lines are processed.

-q qf1, qf2, ... qfn
Quick update mode. Decks that are specified with the -q option and decks specified by a COMPILE directive are written to the compile file. These decks must be in the same case as appeared (uppercase, lowercase, or mixed) in the PL.

You can specify up to 100 names. The maximum size of the string used is 96 characters.

omitted
Normal update mode. Decks specified by COMPILE directives, modified decks, and decks calling modified common decks are written to the compile file.

-o opts

The only option currently available is sq. Other options (available with Cray update) are not currently available.

sq Provides sequence numbers for source and compiler output.

EXAMPLES

The following example shows a modification of a PL:

```
nupdate -p plname -i mods -q DECK3, DECK2, DECK4 -c cpl
```

The options used on this command line indicate the following:

- o The PL file is plname in the current directory.

- o The input is read from the file mods in the current directory.

- o This nupdate run uses quick mode.
  If a single COMPILE directive is used:

```
*COMPILE DECK3,DECK2,DECK4
```

DECK2 through DECK4 are written to the compile file (-c cplfile).

Note: Unlike Cray update the decks are written to the compile file in alphabetical order.

2. Nupdate Directives

2.1 Modification Directives

Modification directives modify text in the Program Library (PL) by adding new lines or changing the active status of existing lines. The changes made by these directives are associated with a modification set. The directives in this category are as follows:

(B)EFORE
(D)ELETE
(I)NSERT
(ID)ENT

2.2 Run Option Directives

/comment
(C)OMPILE
DECLARE
(DEF)INE
(M)ASTER

2.3 Compile Dataset Directives

Compile dataset directives determine the contents and format of the compile dataset. These directives are embedded in the PL as lines that can be added or deleted just the same as text lines. The directives are as follows:

CALL
ELSE
ELSEIF
ENDIF
IF

2.4 Deck Definition Directives

The DECK and COMDECK directives specify whether a new deck is a regular or common deck. These two directives do not fit the categories described in the previous paragraphs. They insert new text into the PL and so are modification directives, but are not associated with modification directives. DECK and COMDECK directives are stored in the PL. Unlike the compile dataset directives, you cannot delete them like text, and you cannot insert text before them.

2.5 Directive Format

A directive has the following format:

```
md p1,p2,p3
```

m Master character.
d Directive name or abbreviated name (must be uppercase).
px Parameter, dependent on directive.

2.5.1 Line Identification

Each modification set and each deck (or common deck) has a unique identifier. This identifier is the name from the corresponding DECK, COMDECK or IDENT directive.

The sequence number for a line from the original deck is derived from the line's position in the deck. New lines inserted by a modification set are sequenced in the order in which they appear in the deck.

A given line is uniquely referred to by id.seq; id is the deck or modification set identifier name and seq is the line sequence number.

The numbering scheme assigns numbers in the order in which the directives are processed.

2.5.2 Identifier Names

Each identifier (deck, common deck or mod set name) is a 1 to 8 character name assigned when the identifier is first used.

2.6 Directives

The set of directives that nupdate recognizes is now described.

In this text a * is used as the master character for descriptions, use of the MASTER directive allows other characters to be used instead for the modification set.

2.6.1 / - comment

A comment, indicated by the comment character /, may be inserted in the modification set and is ignored by nupdate.

Format: */ comment

2.6.2 BEFORE - insert before a line

BEFORE (or B) indicates that lines immediately following it are to be inserted before the line specified (in the PL).

Format: *BEFORE id.seq

id	Deck or modification set identifier name.
seq	Line sequence number.

2.6.3 CALL - call a common deck

CALL indicates the location in which the specified common deck is to be placed when the compile dataset is generated. The combination of common decks and CALL directives lets you maintain a single copy of common text and be assured that the most current copy is always used in a deck that calls it.

Format: *CALL common_deck_name

2.6.4 COMDECK - introduce a common deck

COMDECK introduces a common deck. Lines up to the next DECK, COMDECK, IDENT, INSERT, DELETE, BEFORE, or end of input compose the common deck.

Other directives are interpreted but do not terminate the common deck.

The common deck directive is the first line of the common deck and is assigned sequence number 1.

Format: *COMDECK comdeck_name

2.6.5 COMPILE - specify compile or source dataset

COMPILE (or C) specifies the contents of the compile datasets. In selecting decks for compile output, you need not specify called common decks.

COMPILE directives can occur anywhere in the input, but they must not refer to unknown decks, such as decks introduced later in the same run.

Format: *COMPILE p1[,p2,p3,p4...]

px single deck name

2.6.6 DECK - introduce a deck

DECK introduces a new deck; it is the first line in the deck and has sequence number 1. The deck is composed of lines up to the next DECK, COMDECK, IDENT, INSERT, DELETE or end of input.

Input edit directives and run option directives are interpreted but do not terminate the deck. Decks can contain embedded directives such as CALL.

Format: *DECK deck_name

2.6.7 DEFINE - define names

DEFINE (or DEF) defines a name (1 to 8 characters in length) to be used by an IF directive. Names declared with this directive do not need to be unique from deck or common deck names or modification set identifiers. Defined names are known only in the in the run in which they are defined; they are not stored in the PL.

Format: *DEFINE n1[,n2,n3,n4]

nx defined name.

2.6.8 DELETE - delete lines

DELETE (or D) lets you delete (deactivate) lines or ranges of lines and, optionally, replace them with lines appearing after the DELETE directive. A deletion range must not cross a deck boundary.

Format: Can be any of the following

```
*DELETE id1.seq1,id2.seq2 (range delete)
*DELETE id1.seq1,id2             (range delete, short
form)
*DELETE id1.seq1                 (single line delete)
```

idx deck or modification set identifier name
seqx line sequence number

2.6.9 ELSE - reverse condition

ELSE reverses the condition from the previous IF or ELSEIF directive to determine whether the text following it is written to the compile dataset. You can only use one ELSE with an IF, and ELSE must follow all ELSEIFs associated with that IF.

Format: *ELSE

2.6.10 ELSEIF - test condition

ELSEIF specifies a condition for evaluation when no previous condition in the same IF group was true. If the condition is evaluated as true, ELSEIF writes the text following the directive to the compile dataset, and skips the text following all further ELSEIF and ELSE directives in this IF group. If the condition is false or is not evaluated, it ignores all directives up to the next IF, ELSEIF, ELSE or ENDIF. ELSEIF must have a matching IF, and it cannot follow ELSE; otherwise, an error message is issued.

Format: *ELSEIF type,name[,...,boolean,type,name]

type Type of conditional name, currently only DEF is supported. The name must have been introduced with the DEFINE directive, or the DEF parameter or -d option. A minus sign before the type negates the condition.

name A deck or common deck name, modification set identifier, or defined name, depending on the value of type. Each clause of the condition is true if a name of the proper type is known or, when negated, if the name is unknown.

boolean A logical operator: AND and OR. AND has precedence over OR and is evaluated first.

2.6.11 ENDIF - end conditional text

ENDIF ends a conditional text range and an IF group. Each ENDIF must have a matching IF; otherwise, you will get an error message.

Format: *ENDIF

2.6.12 IDENT - identify modification set

IDENT (or ID) provides the modification set identifier that is to be associated with all of the changes in a modification set. IDENT is the first line in a modification set.

Format: *IDENT ident

ident modification set identifier.

2.6.13 IF - begin conditional text

IF begins a conditional text range and gives the condition under which the range is written to the compile dataset. IF is the beginning of an IF group, which can include ELSEIF and ELSE directives and must end with an ENDIF directive. You can nest IF groups to any level.

If the condition is true, the text following the directive is written to the compile dataset, and the text following all ELSEIF and ELSE directives in the IF group is skipped. If the condition is false, all directives up to the next ELSEIF, ELSE, or ENDIF in the IF group are ignored.

Format: *IF type,name[,...boolean,type,name]

type (See corresponding entry for ELSEIF).

name (See corresponding entry for ELSEIF).

boolean (See corresponding entry for ELSEIF).

2.6.14 INSERT - insert text after a line

INSERT (or I) indicates that the lines immediately following are to be inserted after the line specified.

Format: *INSERT id.seq

id deck or modification set identifier.

seq line sequence number.

2.6.15 MASTER - change input master character

MASTER changes the master character for directives in the input. Directives stored in the PL are not affected by this directive.

Format: *MASTER m

m new master character for directives in the input datasets.