IX.4A-SYSTEM OPERATIONAL FORECAST SYSTEM DATA BASE LOCKS

Bookmarks

Purpose . . . . [Bookmark]
Opening a Set of Lock Files . . . . [Bookmark]
Opening a Lock File . . . . [Bookmark]
Lock Routines . . . . [Bookmark]
Lock Name Creation . . . . [Bookmark]
Tables . . . . [Bookmark]
[Bottom]

Contents

Page
Purpose 1
Opening a Set of Lock Files 2
Opening a Lock File 2
Lock Routines 3
Lock Name Creation 4

[Next] [Previous] [Bookmarks] [Top]

Purpose

An Operational Forecast System (OFS) Data Base lock is a temporary file that indicates that OFS file groups are being used.

The OFS file groups are:

1. PDB: Preprocessor Data Base (PDB*) files
2. PPP: Preprocessor Parametric Data Base (PPP*) files
3. PRD: Processed Data Base (PRD*) files
4. FCESP: Forecast Component and ESP files (FC*, ESP* and RESJ*) files
5. HCLUSER: all other files (USERPARM, HCL*, GLOBAL.*, etc.)

Each of the file groups can be locked or unlocked individually.

Table 1 [Bookmark#1] describes the lock scheme used for the programs that access the OFS data bases.

The result of the locking scheme is that:

- o if the program FCST Function ESP is run with Technique SKIPBLND turned on (i.e. set to 1) it can run simultaneously with programs BATCHPST, SHEFPOST and the program FCST preprocessor Functions
- o program FCST Function FCEXEC can run simultaneously with programs BATCHPST and SHEFPOST

[Back] [Next] [Previous] [Bookmarks] [Top]

Opening a Set of Lock Files

A set of lock files is opened one file at a time. Each lock file is opened for read or write access. If the lock file cannot be opened then all open locks are freed and the program attempts to open the set again starting from the first lock. The number of times an attempt is made to open the set before an error is generated is determined by this Apps_default token:

ofs_locks_max_pass	number of passes made to acquire a set of locks before an error is generated

If the second program cannot open the lock set within the number of passes specified by Apps_default token ofs_locks_max_pass then it should print an error message and stop.

When a program ends all open lock files are closed and the locks are freed.

[Next] [Previous] [Bookmarks] [Top]

Opening a Lock File

A lock file is opened by a program with either read or write access rights.

If a second program tries to open the same lock file when a program already has opened it with write access then the second program waits for a specified amount of time and then tries to open the lock file again. If the second program cannot open the lock file within the specified of time then another an attempt is made to open the set of locks as described in 'Opening a Set of Lock Files' [Bookmark].

When a program finishes it should call routine free_ofs_lock [Bookmark]. If a program ends abnormally or does not free the lock and ends then the lock file will revert to a non-use status since the operating system will free the file when a process dies.

The following Apps_default tokens are used to define the parameters needed for the lock:

locks_dir	directory name to hold lock files
ofs_files	pathname to directory holding OFS file sets
ofs_level	subdirectory under 'ofs_files' that holds the specific file set used by an executable (full pathname: <ofs_files>/<ofs_level>/fs5files)
ofs_fs5files	used as a check for the same full pathname of the file set (same as <ofs_files>/<ofs_level>/fs5files)
ofs_lock_max_wait	maximum number of minutes to wait to get a lock before a status flag indicates that the program should be aborted
ofs_lock_wait_interval ...	number of seconds between retries to get

Lock Routines

The following routines are called to use the locking system:

- o set_ofs_lock which locks all file groups for the program.
- o free_ofs_lock which frees all file groups currently locked by the program
- o hlockfiles which locks file groups for a specified program
- o hunlockfiles which unlocks file groups for a specified program

Routine set_ofs_lock has two arguments: the first is either the character string 'read' or 'write'; the second is the returned status (0 for lock opened, 1 for cannot open because the lock is in use by another program or 3 for cannot open due to a program error).

Routine free_ofs_lock has one argument which is the returned status (0 for no error or 1 for error).

Routine hlockfiles has two arguments; the first is the lock name; the second is the returned status (see set_ofs_lock). Valid lock names:

Program or Function	Mode	Lock Name
BATCHPST		BATCHPST
FCST	non-fcst	NONFCST
FCST	startup	FCST
FCEXEC	no-FFG&ASSIM	FCEXEC_NOFA
FCEXEC	FFG&ASSIM	FCEXEC_FA
ESP	blend	ESP_BLEND
ESP	no blend	ESP_NOBLEND
Preprocessors		PREPROC
SHEFPOST		SHEFPOST
Others		GENERAL

Routine hunlockfiles has in the same two arguments as hlockfiles. The return status is the same that from routine free_ofs_lock.

Routine UPINIO also needs to be called. This routine sets variable UE in block common UPDAIO to the FORTRAN output unit number for status/error messages. It uses the following tokens to set the unit number (set to -1 if no messages are desired):

ofs_error_output	set to 'on' for output to standard error output else set to 'off'
fortran_stderr	set to the FORTRAN standard error unit number

The following is a Fortran example of using the routines set_ofs_lock and free_ofs_lock:

```
   CHARACTER*5  LTYPE
   ...
   CALL UPINIO ()
```

```
...
LTYPE = 'write'
CALL SET_OFS_LOCK (LTYPE,ISTAT)
IF (ISTAT.NE.0) STOP 16
...
CALL FREE_OFS_LOCK (ISTAT)
...
```

[Next] [Previous] [Bookmarks] [Top]

Lock Name Creation

Every lock name has the file group name appended to the end after an underscore ('_'). The acronyms are given in the Table 1 [Bookmark#2].

The lock name is created by first creating two pathnames and then checking if they are the same:

1. Get apps_defaults values for tokens ofs_files and ofs_level and create the pathname <ofs_files>/<ofs_level>/fs5files.

   ```
   Example:  <ofs_files> = /awips/rfc/nwsrfs/ofs/files
             <ofs_level> = ofstest
             pathname = /awips/rfc/nwsrfs/ofs/files/ofstest/fs5files
   ```

2. Get apps_defaults values for token ofs_fs5files as a single complete pathname.

   ```
   Example:  pathname = /awips/rfc/nwsrfs/ofs/files/ofstest/fs5files
   ```

If the pathnames are the same then the lock name is set to ofs.<ofs_level>.

```
Example: lockname = ofs.ofstest_PPP
```

If the pathnames are different (such as an environment variable being reset for one of the tokens in a script) then the lock name is set to the value of token ofs_fs5files with the slashes changed to underscores.

```
Example: lockname = _awips_rfc_nwsrfs_ofs_files_ofstest_fs5files_PPP
```

Table 1. Lock scheme

Program or		------------ File Group ----------				
Function	Mode	PDB	PPP	PRD	FCESP	HCLUSER
BATCHPST		write	NONE	NONE	NONE	NONE
FCST 1/	non-fcst	NONE	NONE	NONE	NONE	write
FCST 1/	startup	NONE	NONE	NONE	NONE	read
FCEXEC 2/	no-FFG&ASSIM	NONE	NONE	write	write	read
FCEXEC 2/	FFG&ASSIM	NONE	read	write	write	read
ESP 3/	blend	NONE	NONE	read	write	read
ESP 3/	no-blend	NONE	NONE	NONE	write	read
Preprocessors 4/		write	read	write	NONE	read
SHEFPOST		write	NONE	NONE	NONE	read
Others		write	write	write	write	write

Notes:

1. The 'non-fcst' and 'startup' modes refers to running program FCST Hydrologic Command Language commands @NONFCST [Hyperlink] and @FCST [Hyperlink]. When program FCST is executed it is initially running as FCST in 'startup' mode.

2. The 'no-FFG&ASSIM' and 'FFG&ASSIM' modes refer to running program FCST Function FCEXEC [Hyperlink] with Techniques FFG [Hyperlink] and ASSIM [Hyperlink] turned off or one of them turned on.

3. The 'blend' and 'no-blend' modes refer to running Function ESP [Hyperlink] with Technique SKIPBLND [Hyperlink] turned on or off.

4. The preprocessors are Functions MAT, MAP, MAPE, MAPX and RRS [Hyperlink]. The Function MAP writes to the Preprocessor Data Base when the Technique WTEST24 [Hyperlink] is turned on.

[Top]