Natural Buffer Pool

This chapter covers the following topics:

- General Information
- Setting up a Buffer Pool
- Using the Utility NATBPSRV for Creating the Buffer Pool
- NATBPSRV Error Messages
- Monitoring the Buffer Pool
- Trouble Shooting
- Shutting Down and Restarting the Buffer Pool

General Information

The Natural buffer pool is used to share Natural objects between several Natural processes that access objects on the same computer. It is a storage area into which compiled Natural programs are placed in preparation for their execution. Programs are moved into and out of the buffer pool as Natural users request Natural objects.

Since Natural generates reentrant Natural object code, it is possible that a single copy of a Natural program can be executed by more than one user at the same time. For this purpose, each object is loaded only once from the system file into the Natural buffer pool, instead of being loaded by every caller of the object.

The following topics are covered below:

- Objects in the Buffer Pool
- Coordination under UNIX
- Multiple Buffer Pools
- Storing Objects in the Buffer Pool
- Read-Only Buffer Pool
- Restrictions

Objects in the Buffer Pool

Objects in the buffer pool can be any executable objects such as programs and maps. The following executable objects are only placed in the buffer pool for compilation purposes: local data areas, parameter data areas and copycodes.

When a Natural object is loaded into the buffer pool, a control block called a directory entry is allocated for that object. This control block contains information such as the name of the object, to which library or application the object belongs, from which database ID and Natural system file number the object was retrieved, and certain statistical information (for example, the number of users who are concurrently executing a program).

Coordination under UNIX

Resource sharing requires that access to the buffer pool be coordinated among all users. Several system resources are necessary to accomplish this. For example, shared memory on the UNIX operating system is used to store the objects and their administrative information. To synchronize access to these objects, a set of semaphores is used. The amount of available shared memory and the number of semaphores is configured statically in the operating system, and as a result, it may be necessary to change system parameters and to recreate the operating system kernel for your installation. Further information about these topics is system-dependent and is described in the installation documentation for your UNIX computer.

Multiple Buffer Pools

Depending on the individual requirements, it is possible to run different buffer pools of the same Natural version simultaneously on the same computer.

Storing Objects in the Buffer Pool

When a user executes a program, a call is made to the buffer pool manager. The directory entries are searched to determine whether the program has already been loaded into the buffer pool. If it does not yet exist in the buffer pool, a copy is retrieved from the appropriate library and loaded into the buffer pool.

When a Natural object is being loaded into the buffer pool, a new directory entry is defined to identify this program, and one or more other Natural objects which are currently not being executed may be deleted from the buffer pool in order to make room for the newly loaded object.

For this purpose, the buffer pool maintains a record of which user is currently using which object, and it detects situations in which a user exits Natural without releasing all its objects. It dynamically deletes unused or out-of-date objects to accommodate new objects belonging to other applications.

Read-Only Buffer Pool

A read-only buffer pool is a special buffer pool that only allows read access. If an object is not found in the read-only buffer pool, Natural issues error 82 (object not found). As no attempt is made to retrieve the missing object in the system files, all lock operations on the system file as well as on the buffer pool are skipped. No account data are gathered. An unlimited number of users can access read-only buffer pool.

A read-only buffer pool is defined in the Configuration Utility (see also *Setting up a Buffer Pool* below). If a buffer pool has been defined as a read-only buffer pool, the values defined for the maximum number of users and for the semaphore key are ignored.

The utility NATBPSRV does not allocate semaphores for a read-only buffer pool. It expects, however, a preload list in a file named <*bufferpool-name*>.*PRL* at the location of the Natural parameter files, which is defined in the local configuration file (installation assignments). For example, if the name of the read-only buffer pool is "ROBP", the file name must be *ROBP.PRL*.

A preload list can be generated using the Natural utility CRTPRL. This utility extracts the contents of a buffer pool and merges it with the existing preload data of a buffer pool.

The preload list in the *PRL* file contains records with comma-separated data in the following form:

```
database-ID,file-number,library,object-name,kind,type
```

The keywords in the file have the same meaning as the keywords shown by the DIR command of the NATBPMON utility.

With the exception of directory-describing records (the kind of object is "D", which means the object is part of *FILEDIR.SAG*), a value must be assigned to all keywords. Examples:

Keywords	NATBPSRV loads the following into the buffer pool
222,111,MY_LIB,PGM1,G,P	Object code of program PGM1 from library MY_LIB which is located on database 222 and file number 111.
222,113,*,*,D	*LIBDIR.SAG* which is located on FNAT=222,113.
222,111,MY_LIB,*,D	*FILEDIR.SAG* from library MY_LIB which is located on FUSER=222,111 .

Using a read-only buffer pool has the disadvantage that the application must be known in detail (as missing objects cannot be loaded). This means that all objects needed by an application must be specified in the preload list. In seldom cases, the complete set of objects needed by an application can be determined in advance.

Secondary Read/Write Buffer Pool

Natural can run with a read-only buffer pool as the primary buffer pool. Such a buffer pool is not modifiable. Objects missing in the read-only buffer pool cannot be loaded. If an object is not found in the read-only buffer pool, Natural issues error 82 (object not found). To avoid this, Natural can attach during execution to a secondary standard buffer pool (which allows read/write access) and activate the missing objects there. If a call to locate an object in the primary buffer pool fails, the secondary buffer pool operates as a backup buffer pool. The dynamic parameter BPID2 identifies the secondary buffer pool.

Other than for the read-only buffer pool, there is a maximum number of users that can attach to the secondary buffer pool and object locking through semaphores takes place each time the secondary buffer pool is accessed.

The preload list of the read-only buffer pool can be updated/enhanced by merging the contents of the secondary read/write buffer pool with the preload list of a read-only buffer pool using the utility CRTPRL.

Alternate Read-Only Buffer Pool

For a read-only buffer pool, it is possible to define the name of an alternate buffer pool in the Configuration Utility (see also *Setting up a Buffer Pool* below).

Using the SWAP command of the NATBPMON utility, which is only available for a read-only buffer pool, you can tag a read-only buffer pool as "obsolete". All Natural sessions attached to an obsolete buffer pool will detach from this buffer pool and will attach to the alternate buffer pool - but only if the alternate buffer pool is also a read-only buffer pool. The swap from one buffer pool to the other occurs either when Natural tries to load a new object (for example, when executing a CALLNAT or RETURN statement) or

when Natural tries to interpret a command which has been put on the stack. The IPC resources (that is, the shared memory segment) of a buffer pool tagged as obsolete can be removed after issuing the SWAP command of the NATBPMON utility. This feature allows exchanging a buffer and its contents by another read-only buffer pool with updated contents without stopping Natural sessions.

Known issues: The IPCRM command of the NATBPMON utility will report an error trying to delete the semaphores associated to a read-only buffer pool.

Creating a Preload List Using the CRTPRL Utility

The Natural utility CRTPRL, which is located in the library SYSBPM, is used to create a preload list for a read-only buffer pool.

The utility uses the content of a source buffer pool as the basis for the preload list and checks whether the preload list already exists for a read-only (target) buffer pool:

- If the preload list exists, the existing data in the preload list is merged with the data from the source buffer pool, and the preload list is saved with the new content.
- If the preload list does not yet exist, it is created using the content from the source buffer pool.

The content of the resulting preload list determines the content of the read-only buffer pool. The preload list is read by the utility NATBPSRV which loads the corresponding objects into the read-only buffer pool.

Restrictions

When using the Natural buffer pool, only minimum restrictions must be considered:

- When a Natural session hangs up, do not terminate it by using the UNIX command kill, the terminal command break or the interrupt key.

  If this session is currently performing changes to the buffer pool internal data structures, an interruption may occur at a stage where the update is not fully completed. If the buffer pool internal data structures are inconsistent, this could have negative effects.

  Note:
  This can only happen when the Natural nucleus is executing buffer pool routines.

- All resources must be shared among all users of one Natural buffer pool. Group membership of a process is used to give access rights for the buffer pool. This means that the shared memory can be changed by all group members, but not by anyone else. The same applies to the semaphores.

  Note:
  All users of the same Natural buffer pool must belong to the same user group on the UNIX operating system.

Setting up a Buffer Pool

The buffer pool assignments are stored in the local configuration file. To set up a buffer pool, you have to specify specific values in the local configuration file using the Configuration Utility. For a list of these values, see *Buffer Pool Assignments* in the *Configuration Utility* documentation.

Using the Utility NATBPSRV for Creating the Buffer Pool

The buffer pool is created using the utility NATBPSRV.

Note:
The utility NATBPSRV should not be accessible to all Natural users, because it can cause damage to the work of other buffer pool users.

NATBPSRV allocates the resources required by the buffer pool and creates the permanent communication facilities (that is, shared memory and semaphores) used for the buffer pool. The necessary specifications for the resources and facilities are made with the Configuration Utility (see *Setting up a Buffer Pool*).

The NATBPSRV utility should only be used during system startup, from within the startup procedure *natstart.bsh*.

By default, the buffer pool NATBP is started. If another buffer pool is to be started, you specify its name with the following NATBPSRV command line option:

```
NATBPSRV BP = buffer-pool-name
```

If NATBPSRV discovers in the process of creating a buffer pool that a buffer pool of the same name is already active, it deletes the already active buffer pool. If the deletion fails, NATBPSRV terminates with an appropriate error message.

NATBPSRV Error Messages

NATBPSRV can issue the following error messages if the buffer pool that is to be created is meant to be a read-only buffer pool:

Unable to attach to buffer pool. Return code ... received from bp_init.

Explanation To load the objects described in the preload list, NATBPSRV attaches to the previously created buffer pool as a user. The attach process failed.

Action Contact Software AG Technical Support.

Unable to get parameter path.

Explanation The path defined in the local configuration file identifying Natural’s parameter files could not be established.

Action Contact Software AG Technical Support.

File ... is not accessible.

Explanation The preload list is not accessible or not present.

Action Revise access rights or create a preload list.

Unable to open file ...

Explanation The preload list cannot be read.

Action Re-create preload list.

Skipped erroneous record: '...'. Buffer pool may not operate correctly.

Explanation An invalid record was found in the preload list. The record is skipped and the load process is continued. There may arise errors in your application due to missing objects.

Action Correct the record if it has been created manually, or contact Software AG Technical Support.

Unable to retrieve LIBDIR.SAG in FNAT(...,...). Application will not run.

Explanation *LIBDIR.SAG* was not found. An application depending on `FNAT(...,...)` will not run.

Action Correct the record if it has been created manually, or contact Software AG Technical Support.

Buffer pool manager returned with error code ... . Buffer pool is not operational.

Explanation *FILEDIR.SAG* could not be loaded into the buffer pool. The buffer pool is either too small to hold *FILEDIR.SAG* or *FILEDIR.SAG* is damaged. The previously listed message tells which *FILEDIR.SAG* is causing the trouble.

Action Correct the record if it has been created manually, or contact Software AG Technical Support.

Buffer pool manager returned with error code ... . Error ... occurred.

Explanation An error occurred loading an object into the buffer pool.

Action Normally, the size of the buffer pool is too small. Increase its size and repeat the operation. If the problem remains, contact Software AG Technical Support.

Object ... in library ... on system file (...,...) not found. Application may not run.

Explanation The preload record processed pointed to an object that was not found. This normally happens if an application is modified and the corresponding preload list is not updated.

Action Remove/revise preload record in question

Preload executed. Buffer pool is ready to run.

Explanation All preload records were processed. The buffer pool is unlocked and Natural can access that buffer pool.

Monitoring the Buffer Pool

The Buffer Pool Monitor is used to oversee the buffer pool's activity during its operation. The Buffer Pool Monitor can also be used to shut down the buffer pool when Natural must be stopped on a computer.

The Buffer Pool Monitor collects information on the current state of your Natural buffer pool.

If multiple buffer pools are active on the same computer and an object that is loaded to more than one buffer pool is modified by a Natural process, the object will only be removed from the buffer pool to which the modifying Natural process is attached.

For detailed information for how to use the Buffer Pool Monitor, see *Using the Buffer Pool Monitor (NATBPMON)*.

Trouble Shooting

This section describes problems that may occur when using the Natural buffer pool and how to solve them.

It is assumed that you are familiar with the UNIX commands `ipcs` and `adb`.

The following are typical command output examples, with an explanation of what went wrong during execution.

Problem 1

Either Natural or the Natural Buffer Pool Monitor (`NATBPMON` utility) cannot be started.

Examples

The following examples describe the most typical problems you are likely to encounter as a Natural administrator or user. These problems occur when you start Natural or the Natural Buffer Pool Monitor, and the buffer pool is not active.

- You try to start Natural with the following command:

```
natural bp = sag
```

The following message appears:

```
Natural Startup Error:    16
Unable to open Buffer Pool,
Buffer Pool error:   "unexpected system call error occurred "  (20)
Global shared memory could not be attached.:   shmkey = 11111111

Operating System Error 2 - No such file or directory
```

- You try to start the Natural Buffer Pool Monitor with the following command:

```
natbpmon bp = sag
```

When you enter the WHO command at the NATBPMON prompt, the following message appears:

```
Buffer Pool error:   unexpected system call error occurred  (20)
Global shared memory could not be attached.:   shmkey = 11111111
Operating System Error 2 - No such file or directory
```

Solution

1. Start the buffer pool service as described in *Using the Utility NATBPSRV for Creating the Buffer Pool*.
2. Use the UNIX command ipcs to verify the existence of the necessary semaphores and the shared memory:

```
ipcs -m -s
```

This results in the following output:

```
IPC status from /dev/kmem as of Mon 23-MAY-2005 12:03:24.30
T     ID     KEY        MODE       OWNER    GROUP
Shared Memory:
m    807 0x4e425031 --rw-rw----      sag  natural
Semaphores:
s     85 0x4e425031 --ra-ra----      sag  natural
```

Note:
The above output was edited to exclude memory segments and semaphores that do not belong to the Natural buffer pool.

If you cannot find a shared memory segment or a set of semaphores with the key you assigned them, the buffer pool was not started.

Problem 2

The Natural buffer pool and a Natural utility are not of the same Natural version.

Examples

If a utility tries to use the buffer pool, the utility and buffer pool versions are checked for equality. If they differ, the access is denied and an error message is output.

- You try to start Natural with the following command:

  ```
  natural
  ```

  The following message appears:

  ```
  Unable to open buffer pool,
  contact your system administrator
  bp_error: 25, version mismatch of buffer pool
  ```

- You try to start the Natural Buffer Pool Monitor (NATBPMON utility) with the following command:

  ```
  natbpmon
  ```

  When you enter the DIR command at the NATBPMON prompt, the following message appears:

  ```
  bp_init: res = -1, bp_errno = 25, errno = 0
  buffer pool error message: "version mismatch of buffer pool"
  ```

Solution

Verify that all utility programs used with the buffer pool are of the same Natural version.

1. To ascertain the versions, use the UNIX command adb:

   ```
   adb $NATDIR/$NATVERS/bin/natural
   ```

2. Enter the following:

   ```
   bp_majrel?D
   ```

   This results in the following output:

   ```
   natural'bp_majrel:
   natural'bp_majrel:    1
   ```

3. Enter the following:

```
bp_minrel?D
```

   This results in the following output:

```
natural'bp_minrel:
natural'bp_minrel:    3
```

4. Enter the following:

```
bp_version?D
```

   This results in the following output:

```
natural'bp_version:
natural'bp_version:   435
```

   The output from the above commands identifies a buffer pool for Version 1.3, which has the sequence version 435.

   If all programs are of the same Natural version, but contain different buffer pool versions, contact Software AG Support.

5. Press CTRL+D.

Note:
The above commands also work with the images NATBPMON and NATBPSRV.

Shutting Down and Restarting the Buffer Pool

Usually it should not be necessary to shut down and restart the buffer pool. This may only be necessary if the buffer pool should become unusable due to serious internal errors in the buffer pool, which is extremely unlikely to occur, or because the parameters defining the buffer pool structure became obsolete.

If the NATBPMON utility is still able to access the buffer pool, proceed as follows:

1. Shut down the buffer pool with the SHUTDOWN command of the NATBPMON utility.

   Once the SHUTDOWN command is executed, new users are denied access to the buffer pool.

   Tip:
   Active buffer pool users can be monitored by issuing the WHO and STATUS commands of the NATBPMON utility.

2. After the last user has stopped accessing the buffer pool, buffer pool resources can be deleted by issuing the IPCRM command of the NATBPMON utility.

3. To restart the buffer pool, call the file *natstart.bsh* from a sufficiently privileged account.

If you have super user rights, you can use the FORCE option of the SHUTDOWN command:

1. Shut down the buffer pool with the SHUTDOWN FORCE *grace-period* command of the NATBPMON utility.

   This command - like the SHUTDOWN command without options - denies new users access the buffer pool. However, the terminate signal SIGTERM is sent to all active Natural sessions, forcing them to log off from the buffer pool.

   If the optional parameter *grace-period* is omitted, this command waits until all active sessions have performed their shutdown processing and then executes the IPCRM command of the NATBPMON utility .

   If the optional parameter *grace-period* has been specified, NATBPMON waits the specified number of seconds before it executes its IPCRM command - regardless of the closedown status of the sessions logged on to the buffer pool. Therefore, the value defined for the grace period should be sufficiently large to allow the sessions to terminate in time.

   Note:
   SHUTDOWN FORCE 0 is the same as SHUTDOWN FORCE (without the parameter *grace-period*).

2. To restart the buffer pool after successful execution of the SHUTDOWN FORCE command, call the file *natstart.bsh* from a sufficiently privileged account.

If the NATBPMON utility is not able to perform a clean shutdown of the buffer pool, the buffer pool must be deleted by using operating system commands:

1. Use the UNIX command ipcs to find out the status of the buffer pool's shared memory and semaphores:

   ```
   ipcs -a -m
   ```

   In the column **NATTCH** of the output of an ipcs -m -a command, you can see the number of processes currently attached to a shared memory segment. For example:

   ```
   IPC status from /dev/kmem as of Mon May 23 12:15:38.39 2002
   T         ID     KEY        ...  OWNER  GROUP  ... NATTCH    SEGSZ
   Shared Memory:
   m        707 0x4e425031  ... sag    natural ...      7   153600
   ```

2. It is highly probable that the number of processes attached to shared memory incorporates a Natural nucleus or the NATBPMON utility currently running. Inform the users who run these processes and ask them to terminate their sessions or terminate them yourself by using the UNIX command kill once you have found out their process IDs using the ps command.

3. Once you are sure that no one is using the buffer pool for important work, its resources can be deleted by using the UNIX command ipcrm. For example:

```
ipcrm -M 0x4e425031 -S 0x4e425031
```

The values specified for the `-M` and `-S` options must be those that were specified inside the parameter file used to start the buffer pool.

Be careful when you delete shared memory and semaphores using the UNIX command `ipcrm`. If you accidentally delete the wrong resource, this might have a serious impact on other software products running on your computer.

4. The result of deletion can be verified by using the UNIX command `ipcs` again.

   If there are still some memory segments or message queues displayed, they could belong to other software, or they are marked for deletion because some other process is still attached to them.

If the buffer pool cannot be started after removing the shared memory and semaphores, you should consider either rebooting your computer or contacting Software AG Support.