

Natural Buffer Pool

This chapter covers the following topics:

- General Information
 - Setting up a Buffer Pool
 - Using the Natural Buffer Pool Service
 - Using the Utility NATBPSRV for Creating the Buffer Pool
 - Monitoring the Buffer Pool
 - Trouble Shooting
 - The Natural Client Buffer Pool Service
-

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
- Resource Handling
- 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 dialogs. 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).

Resource Handling

Resources are loaded into the buffer pool if they reside in a library of a Natural system file (for example, FUSER) and if their names do not exceed 32 characters (including the file extension).

Each resource that resides in the directory which is assigned to the environment variable NATGUI_BMP or whose name is longer than 32 characters, is loaded directly into the Natural process every time it is accessed (that is: the resource is not loaded into the buffer pool).

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.

For each buffer pool, synchronization can be enabled in the Configuration Utility (see also *Setting up a Buffer Pool* below). All buffer pools which contain objects from the same system file and for which synchronization has been enabled are then synchronized automatically.

Important:

If the system file resides on a shared drive, synchronization only works if the file system on the server is NTFS.

The following applies when synchronization has been enabled: If an object that is loaded to more than one buffer pool is modified by one Natural process, it is first marked as invalid. When the object is no longer used by any process, it is deleted from the buffer pool. The next time this object is requested by a process, it will be loaded into the buffer pool again.

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 value defined for the maximum number of users is 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	<i>LIBDIR.SAG</i> which is located on FNAT=222, 113.
222, 111, MY_LIB, *, D	<i>FILEDIR.SAG</i> 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.

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 Windows Task Manager.

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.

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 Natural Buffer Pool Service

Natural uses a Windows service, the **Software AG Natural *n.n* Buffer Pool Service**, to start the Buffer Pool Server when the PC is booted.

Natural is installed with the default buffer pool NATBP. NATBP is also used as the default buffer pool name at Natural startup (a different buffer pool can be defined using the profile parameter BPID).

You can modify the service configuration to meet your requirements. This is explained in the following topics:


- Buffer Pool Service Commands
- Example: Starting Natural with Your Own Buffer Pool

Buffer Pool Service Commands

The file *natbpsvc.exe*, which is stored in the Natural *bin* directory, is used to execute the service commands.

The following service commands can be specified in the **Command Prompt** window of Windows:

Command	Description
NATBPSVC INSTALL <i>mode</i>	<p>Installs the buffer pool service. <i>mode</i> can be one of the following:</p> <p>manual Default. The service is installed and must be started manually (either with the START command or by starting the Software AG Natural <i>n.n</i> Buffer Pool Service in Windows).</p> <p>automatic The service is installed and is automatically started when the PC is booted.</p>
NATBPSVC CREATE <i>buffer-pool-name</i>	Creates a new buffer pool to be started by the service. The service checks whether the buffer pool with the specified name is defined in the Natural parameter file.

Command	Description
NATBPSVC START	Starts the service (if not yet active) and all created buffer pools (see the CREATE command) for which the start parameter has been set to "yes" (see the SET command).
NATBPSVC START <i>buffer-pool-name</i>	Starts the specified buffer pool. If the service has not been started (either automatically at boot time or manually by the user) an error message appears.
NATBPSVC SET <i>buffer-pool-name</i> start= <i>mode</i>	Defines whether the specified buffer pool is to be started when the service is started. <i>mode</i> can be one of the following: yes The buffer pool is started. no Default. The buffer pool is not started.
NATBPSVC STOP	Stops the service and all previously started buffer pools.
NATBPSVC STOP <i>buffer-pool-name</i>	Stops the specified buffer pool.
NATBPSVC SHOW	Displays the configuration parameters for all buffer pools that are defined for the service, that is: whether these buffer pools are to be started when the service is started.
NATBPSVC SHOW <i>buffer-pool-name</i>	Displays the configuration parameters for the specified buffer pool, that is: whether this buffer pool is to be started when the service is started.
NATBPSVC STATUS	Displays the status of all buffer pools that are defined for the service, that is: whether these buffer pools are active or not active.
NATBPSVC STATUS <i>buffer-pool-name</i>	Displays the status of the specified buffer pool, that is: whether this buffer pool is active or not active.
NATBPSVC DELETE <i>buffer-pool-name</i>	Deletes the specified buffer pool from the service.  Warning: Do not delete the default buffer pool NATBP, as it is possible that Natural may not function properly anymore.
NATBPSVC REMOVE	Removes the service from the system.

Example: Starting Natural with Your Own Buffer Pool

This example explains how to create a new buffer pool with the name MYBP and how to start Natural with your new buffer pool.

▶ To start Natural with your own buffer pool

1. Use the Configuration Utility to define an additional buffer pool with the name MYBP in the local configuration file. See *Buffer Pool Assignments* in the *Configuration Utility* documentation.

Note:

For this example, you can use the same values as defined for the default buffer pool NATBP.

2. Invoke the **Command Prompt** window of Windows.
3. Go to the Natural *bin* directory which contains the file *natbpsvc.exe*.
4. Enter the following command to create a buffer pool with the name MYBP:

```
NATBPSVC CREATE MYBP
```

The following information is shown:

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: New buffer pool 'MYBP' created
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

Important:

When the buffer pool with the specified name has not yet been defined in the local configuration file, a corresponding message is shown instead. Make sure to define the buffer pool in the local configuration file before you proceed with the steps below.

5. Enter the following command to define that your buffer pool is to be started when the service is started:

```
NATBPSVC SET MYBP start=yes
```

The following information is shown:

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Configuration successfully set
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

6. Enter the following command to start your buffer pool now (without having to restart the service):

```
NATBPSVC START MYBP
```

The following information is shown:

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Send request to Natural n.n Buffer Pool Service
%NATBPSVC-I: Buffer pool 'MYBP' started
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

7. Enter the following command to display the status of all buffer pools that are currently defined for the service:

```
NATBPSVC STATUS
```

The following information is shown:

```
%NATBPSVC-I: Natural n.n Buffer Pool Service
%NATBPSVC-I: Send request to Natural n.n Buffer Pool Service
%NATBPSVC-I: MYBP is active
                NATBP is active
%NATBPSVC-I: Natural n.n Buffer Pool Service
```

8. Start Natural with the dynamic parameter BPID as shown below:

```
natural BPID=MYBP
```

Using the Utility NATBPSRV for Creating the Buffer Pool

The buffer pool is created using the utility NATBPSRV.

The buffer pool server is automatically started by the Natural Buffer Pool Service.

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*).

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
```


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. To ensure that modified objects are also removed from other buffer pools on the same computer to which the object is currently loaded, you can enable the buffer pool synchronization in the Configuration Utility.

Natural provides two versions of the Buffer Pool Monitor: a graphical user interface and the NATBPMON utility which is a command line version.

For detailed information for how to use the different versions of the Buffer Pool Monitor, see *Using the GUI Version of the Buffer Pool Monitor* and *Using the Command Line Version of 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.

The following is a typical command output example, with an explanation of what went wrong during execution.

Problem

Either Natural or the Natural Buffer Pool Monitor 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 NATBPMON utility, and the buffer pool is not active.

- You try to start Natural and the following message appears:

```
Natural Startup Error 16: Unable to open buffer pool.  
Buffer pool error: "unexpected system call error occurred" (20)  
Buffer pool could not attach to the global shared memory.
```

- You try to start the Natural Buffer Pool Monitor and the following message appears:

```
Cannot get shared memory  
Buffer pool error: "unexpected system call error occurred" (20)  
Buffer pool could not attach to the global shared memory.
```

Solution

Start the buffer pool service as described in *Using the Natural Buffer Pool Service*.

The Natural Client Buffer Pool Service

When Natural or Natural Runtime is installed on a file server, a service called **Software AG Natural *n.n* Client Buffer Pool Service** is installed in the client environment (*natchbpsvc.exe*). This service is only installed during a client installation. It is used to create the shared memory for the Natural buffer pool server.

In the client environment, the Natural buffer pool server is started by a Natural process. This is different from a local installation on a PC where the Natural buffer pool server is started by the Natural buffer pool service.

The client buffer pool service cannot be configured using service commands.