

SYSBPM - Buffer Pool Statistics

This function invokes the buffer pool statistics submenu. From the submenu, you can start buffer pool related, object independent statistics functions including hash table statistics.

To select the statistics submenu

- enter code "A" in the code field of the SYSBPM Main Menu, or enter the direct command DISPLAY STATISTICS.

The Buffer Pool Statistics Main Menu is then displayed, from which you can select the following functions:

- General Buffer Pool Statistics
- Buffer Pool Load/Locate Statistics
- Buffer Pool Fragmentation
- Internal Function Usage
- Buffer Pool Hash Table Statistics

General Buffer Pool Statistics

This function is used to monitor the performance of the buffer pool, and displays various statistics regarding the activity of the buffer pool.

You can either select this function on the Buffer Pool Statistics Main Menu, or invoke it by entering the direct command DISPLAY GENERAL.

The "General Buffer Pool Statistics" screen is then displayed. The statistics displayed on this screen are "snapshots" of the buffer pool which are refreshed each time you press ENTER. The following information is displayed:

Item	Explanation
Buffer Pool Address	Shows the storage address of the buffer pool; that is, of the buffer pool control block.
Directory Section	Shows the storage address of the buffer pool directory section (relative to the beginning of the buffer pool). Each object stored in the buffer pool requires a directory entry that contains information about this object. The space for these directory entries is acquired from the buffer pool itself.
Text Record Section	Shows the storage address of the text record section (relative to the beginning of the buffer pool). After the space used by the directory entries has been allocated, the remaining space is divided into blocks called text records (whose size, by default, is 4 KB). A Natural object can occupy one or more text records, depending on its size.
Dataspace attached	Shows the name of the attached dataspace for the BP cache.
Buffer Pool Size (MB)	Shows the size of the whole buffer pool in MB. The buffer pool size can be changed with the NTBPI macro in the parameter module or with the BPI profile parameter (as described in Natural Operations for Mainframes).
Directory Size	Shows the size of a directory entry in bytes.

Item	Explanation
Text Record Size	<p>Shows the size of a text record in KB. The text record size can be changed with the NTBPI macro in the parameter module or with the BPI profile parameter (as described in Natural Operations for Mainframes).</p> <p>The default text record size is set to 4 KB; however, if you use applications that consist of many rather small objects, it is recommended that you reduce it to 2 KB. This reduces the percentage of unused space in the buffer pool, although it can lead to search algorithm 2 being invoked more frequently.</p>
Initialization	Shows the date and time the buffer pool was initialized.
Last Refresh	Shows the date and time the buffer pool was most recently refreshed, and the ID of the user who performed the refresh.
Text Records - Total	Shows the total number of text records.
Text Records - Used	Shows the number of text records currently used.
Text Records - Used (%)	Shows the percentage of currently used text records.
Text Records - Max Used	Shows the maximum number of used text records.
Text Record - Total Size	<p>Shows the total text record size, which is the number of active text records times the single text record size.</p> <p>The difference between the Total Text Record Size and the Total Object Size shows the amount of unused size in the text record section and can also be an indicator for the system administrator of whether to modify the text record size or not.</p>
Text Record - Avg Usage (%)	<p>Shows the average usage of text records in percent.</p> <p>This value should not be significantly less than 75%. If the buffer pool is almost full, any value above 75% indicates a good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.</p>
Space Used (%)	<p>Shows the actual usage of the buffer pool space in percent.</p> <p>Note: If the buffer pool is almost full (that is, the value in the Text Records Used field - see above - is almost 100%), any value above 75% indicates a good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.</p>
Objects - Loaded	Shows the number of objects currently loaded in the buffer pool.
Objects - Max Loaded	Shows the maximum number of objects loaded in the buffer pool.
Objects - Total Size	Shows the total size of the objects currently loaded in bytes.
Objects - Avg TR Used	Shows average number of text records used by one object.
SumOfUseCounts	Sums up the use counts of all objects currently loaded in the buffer pool. A use count counts all current users of a given object. If an object is currently not in use, its use count returns to 0.
AvgLifetimeUsed (min)	Shows the average load time (in min) of objects currently loaded in the buffer pool.
AvgLifetimeReplace (min)	Shows the average load time (in min) of objects, which have already been replaced, i.e. deleted in the buffer pool.

Buffer Pool Load/Locate Statistics

This function provides statistical information on the loading of objects into the buffer pool and the locating of objects in the buffer pool. This information also serves as buffer pool performance indicator.

You can either select this function on the Buffer Pool Statistics Main Menu, or invoke it by entering the direct command DISPLAY LOAD.

The "Buffer Pool Load/Locate Statistics" screen is then displayed. The statistics displayed on this screen are "snapshots" of the buffer pool which are refreshed every time you press ENTER. The following information is displayed:

Information	Explanation
Total Locate Calls	Shows the total number of program location calls; that is, the total number of times the Natural buffer pool manager was requested to search the buffer pool for an object. If the location is successful, the object has been loaded from the buffer pool or the cache and need not be loaded from a Natural system file thereby saving calls and I/Os.
Total Locate Calls successful	Shows the total number of successfully performed locate calls in absolute figures.
Total Locate Calls failed	Shows the total number of locate calls that failed.
Quick Locates	Shows the total number of quick locate requests. Quick location means that the directory address of the last call of the requested program is still available. This is due to the fact that Natural maintains user-specific tables of internal directory entries which contain information about the most recently used objects by each Natural user. When a user invokes an object that has been used before in the Natural session, Natural passes this information to the buffer pool manager, which then bypasses the normal locate procedure. If the last call address cannot be found, a normal locate request is automatically scheduled by the buffer pool manager.
Quick Locates successful	Shows the number of quick locate requests which have been successfully performed.
Quick Locates failed	Shows the number of quick locate requests which failed. Failed quick locates result in Normal Locate calls.
Normal after Quick	Shows the number of normal locate calls that have been preceded by a quick locate call.
Normal after Quick successful	Shows the number of normal locate requests that have been successful in locating the required Natural object in the buffer pool or the cache and have been preceded by a quick locate call.
Normal after Quick failed	Shows the number of normal locate calls that failed and have been preceded by a quick locate call.
Normal Locates	Shows the total number of normal locate requests.
Normal Locates successful	Shows the number of normal locate requests that have been successful in locating the required Natural object in the buffer pool.
Normal Locates failed	Shows the number of normal locate requests which failed. A failed normal locate request indicates that a program has to be loaded from the database or from the buffer cache.

Information	Explanation
Locates from cache	Shows the total number of successful locates of objects that resided in the buffer pool cache. This information is counted only if the previous Locate ("Normal after Quick failed" or "Normal Locates failed") failed. It indicates the number of database loads saved. This means, that, without the buffer pool cache, the object would have to be loaded from the database.
Load Calls (DB)	Shows the total number of load calls made since the buffer pool has been refreshed.
Program Loads	Shows the number of times a Natural object was loaded from a Natural system file into the buffer pool. As several load calls may be necessary to load a single object, this value provides the actual number of program loads made since the most recent buffer pool refresh.
Finished Loads	Shows how many of the program loads above have been successfully executed. For example, a load call could have failed, because at the time of the load, all objects in the buffer pool were currently being executed. When loading a Natural object, the buffer pool manager uses two search algorithms (see below).
Concurrent Loads	Shows the number of loads that have been done to load an object, which is being loaded already by another application. This means, that the load of an object is not yet finished, but another application requested the same object and the object will be loaded more than once.
Algorithm 1	Shows the number of times a storage allocation request was satisfied by Search Algorithm 1. The storage allocation request may be triggered either by a load from the database or by a load from the buffer pool cache. Search Algorithm 1 attempts to find a single piece of storage in the buffer pool or a single object that is not currently being executed and can be deleted, in order to obtain the required space to accomplish the load. If an object has to be deleted, Algorithm 1 compares the time stamps of the objects in question and the oldest one will be deleted. The search for space begins at the top of the text record section. If storage is unavailable after Algorithm 1 completes its search, Algorithm 2 is invoked.
Algorithm 2	Shows the number of times a storage allocation request was satisfied by Search Algorithm 2. The storage allocation request may be triggered either by a load from the database or by a load from the buffer pool cache. Search Algorithm 2 is a more expensive search than is Algorithm 1. Algorithm 2 attempts to combine two or more entities (free storage and/or objects not being executed) in order to obtain the necessary storage for the load. The ages of the objects, however, are not taken into account. Algorithm 2 begins its search at a point in the text record section where it last left off doing such a search, continues to the bottom of the text record section and, if necessary, wraps around to the top of the text record section to make one final pass from the top to the bottom. If space is still unavailable, Algorithm 2 fails, the object cannot be loaded and a corresponding error message is returned.
Largest Allocation	Shows the largest single allocation size so far requested, specified in number of text records.
Allocation Failure	Shows the total number of times an object load failed. The reason for a failure is that either all directory entries are in use at the time of the load request or not enough storage is available in the text record section to perform the load.
Last Failing Sizes	Shows the number of text records that would have been required by the three most recent storage allocation requests that failed. Storage allocation request failures are a direct result of search algorithm 2 failing for an object load request.

Information	Explanation
Total Locate Calls successful/Program Loads	This statistic is expressed as a ratio using "Total Locate Calls successful" and "Program Loads". A value greater than 1 indicates that Natural located more objects in the buffer pool than it loaded from the system file. This ratio serves as a buffer pool efficiency indicator. The larger the number, the better the buffer pool is performing. This is the <i>primary indicator</i> of performance from one buffer pool session to the next.

Buffer Pool Fragmentation

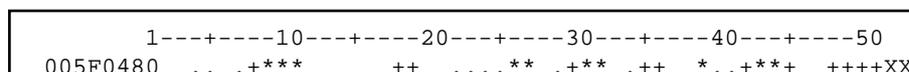
This function provides an overview of the buffer pool fragmentation; that is, an overview of how many different objects occupy how many text records, and how the object locations are spread over the buffer pool.

You can either select this function on the Buffer Pool Statistics Main Menu, or invoke it by entering the direct command DISPLAY FRAGMENTATION.

The "Buffer Pool Fragmentation" screen is then displayed. It provides the following information:

- Buffer Pool Size
- Buffer Pool Address
- Text Record Section
- Text Record Size
- Number of Text Records

In addition, the screen displays a diagram which shows how many different individual objects occupy how much text record size. For example:



Each symbol in the diagram represents one text record, and each sequence of equal symbols represents a different individual object occupying one or more text records. The symbols have the following meaning:

_ and .	Objects with a use count of 0.
+ and *	Objects with a use count greater than 0.
<i>blank</i>	An unused text record.
X	The end of the buffer pool, which means that no further text records are available.

In the example above, the buffer pool contains 48 text records. Three of them are not in use; the rest is occupied by 24 different Natural objects, 12 of them with a use count of 0, and 12 with a use count greater than 0.

Internal Function Usage

This function provides statistical information on the calls to the Natural buffer pool manager.

You can either select this function on the Buffer Pool Statistics Main Menu, or invoke it by entering the direct command DISPLAY FUNCTION.

The "Internal Function Usage" screen then is displayed.

The statistics displayed on that screen are "snapshots" of the buffer pool which are refreshed every time you press ENTER.

The field "Total Calls" shows the overall number of all internal calls that have been made to the buffer pool manager.

Internally, the buffer pool manager can be invoked for various different functions. For each function, the number of times it has been invoked is displayed, both as an absolute figure and as a percentage figure. In addition, these figures are represented in a horizontal bar chart.

Buffer Pool Hash Table Statistics

This function displays statistics about hash table slots and collisions per slot. The statistics determine the efficiency of the hash algorithm used.

You can either select this function on the Buffer Pool Statistics submenu, or invoke it by entering the direct command DISPLAY HASH.

The "Hash Table Collisions" screen is then displayed. The statistics displayed on this screen are "snapshots" of the hash table which are taken every time you press ENTER. The following information is displayed:

Information	Explanation
Total number of slots	Shows the total number of hash table slots; that is, the total possible entries that link the object name with the location of the object. The number of slots i.e. the size of the hash table will be calculated internally depending on the number of text records.
Number of slots used	Shows the number of slots that have one or more entries.
Number of slots free	Shows the number of slots that have no entry.
Max. Collisions per slot	Shows maximum number of collisions of all slots. The maximum number of collisions is the longest possible search path for an object.
Collisions	The number of possible collisions. 0 means no collision or one entry. When there are more than 5 collisions, the number of collisions will be specified in ranges (e.g. 6 - 10).
Number of Slots	Shows the number of slots grouped by their number of collisions. For example, if the number of collisions is 3, the search algorithm must side step a maximum of 3 times to find an object. In addition the percentage of these slots related to all slots used is displayed.
Totalled Number of Slots	Shows the same values as Number of Slots, but the values are summed up.