

# SYSBPM - Buffer Pool Statistics

This function invokes the Buffer Pool Statistics main menu, which is used to obtain buffer-pool-related statistics (including hash table statistics) that are independent of Natural programming objects.

## To invoke Buffer Pools Statistics

- On the SYSBPM Main Menu, enter Function Code **A**.  
Or, in the command line, enter DISPLAY STATISTICS.

The Buffer Pool Statistics main menu is displayed.

The Buffer Pool Statistics main menu provides 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 statistics regarding the activity of the buffer pool.

### To invoke General Buffer Pool Statistics

- On the Buffer Pool Statistics main menu, enter Function Code **G**.  
Or, in the command line, enter DISPLAY GENERAL.

The General Buffer Pool Statistics screen is displayed.

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

Field	Explanation
Buffer Pool Address	The address of the buffer pool.
Directory Section	The address of the buffer pool directory section relative to the beginning of the buffer pool. Each object loaded in the buffer pool requires a directory entry that contains information on this object. The space for these directory entries is allocated in the buffer pool.
Text Record Section	The 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). An object can occupy one or more text records, depending on its size.
Dataspace attached	The name of the dataspace (BP cache) attached to the buffer pool.

Field	Explanation
Buffer Pool Size (MB)	<p>The size of the whole buffer pool in MB.</p> <p>The buffer pool size can be specified with the NTBPI macro in the parameter module or with the BPI profile parameter as described in the Natural Parameter Reference documentation.</p>
Directory Entry Size	The size of a directory entry in bytes.
Text Record Size (KB)	<p>The size of a text record in KB. The text record size can be specified with the NTBPI macro in the parameter module or with the BPI profile parameter, as described in the Natural Parameter Reference documentation. You can change the text record size of an existing buffer pool if you reinitialize the buffer pool by using the INITIALIZE command.</p> <p>The default text record size is set to 4 KB. However, if you use applications that consist of many rather small objects, we recommend that you reduce it to 2 KB. This reduces the percentage of unused space in the buffer pool, although it can lead to Algorithm 2 (see METHOD=S in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation) being invoked more frequently.</p>
Buffer Pool Start	The date and time when the buffer pool was originally started.
Last Initialization	<p>The date and time when the buffer pool was most recently initialized, and the ID of the user who performed the initialization.</p> <p>The buffer pool is initialized when:</p> <ul style="list-style-type: none"> <li>● originally starting the buffer pool,</li> <li>● executing the INITIALIZE command of the SYSBPM utility, or</li> <li>● executing the REFRESH function of the GBP operating program (see Global Buffer Pool Operating Functions in the section Natural Buffer Pools in the Natural Operations documentation).</li> </ul>
Text Records - Total	The total number of text records.
Text Records - Used	The number of text records currently used.
Text Records - Used in %	The percentage of text records currently used.
Text Records - Max Used	The maximum number of text records used.
Text Records - Total Size	<p>The total space used by all text records used, which is Text Records Used multiplied by the size of a single text record.</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 Records - Avg Usage %	<p>The average usage in percent of all text records used, which is Objects - Total Size divided by Text Records - Total Size.</p> <p>This value should not be significantly less than 75%. If the buffer pool is almost full, any value above 75% indicates good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.</p>

Field	Explanation
Space Used %	The actual usage in percent of the text record section, which is Objects - Total Size divided by the total size of the Text Records section.  Note: If the buffer pool is almost full (that is, the value in the field Text Records - Used is almost 100%), any value above 75% indicates good usage of the buffer pool. If the usage is significantly less than 75%, the text record size should be reduced.
Objects - Loaded	The number of objects currently loaded in the buffer pool.
Objects - Max Loaded	The maximum number of objects ever loaded simultaneously in the buffer pool since the buffer pool was started.
Objects - Total Size	The total size in bytes of the objects currently loaded.
Objects - Avg TR Used	The average number of text records used by one object.
Objects - SumOfUseCounts	Totals the Use Counts of all objects currently loaded in the buffer pool.  The Use Count counts all applications currently executing an object. If an object is currently not in use, its Use Count returns to <b>0</b> (zero).
Objects - AvgLifetimeUsed (min)	The average life time (in minutes) of objects currently loaded in the buffer pool.
Objects - AvgLifetimeReplace (min)	The average life time (in minutes) of objects, which have already been replaced, that is 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 an indicator of buffer pool performance.

### To invoke Buffer Pool Load/Locate Statistics

- On the Buffer Pool Statistics main menu, enter Function Code **L**.  
Or, in the command line, enter **DISPLAY LOAD**.

The Buffer Pool Load/Locate Statistics screen is displayed.

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

Field	Explanation
Total Locate Calls	The total number of object 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 found in the buffer pool or the BP cache and need not be loaded from a Natural system file thereby saving calls and I/Os.
Total Locate Calls - successful	The total number of successful Locate calls as an absolute number.
Total Locate Calls - failed	The total number of Locate calls that failed.

<b>Field</b>	<b>Explanation</b>
Quick Locate Calls	<p>The total number of quick Locate calls.</p> <p>Quick location means that the directory address of the last call of the requested object is still available. This is due to the fact that Natural maintains user-specific tables of internal directory entries which contain information on the objects used most recently 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 call is automatically scheduled by the buffer pool manager.</p>
Quick Locate Calls - successful	The number of quick Locate calls that have been successfully performed.
Quick Locate Calls - failed	<p>The number of quick Locate calls that failed.</p> <p>Failed quick Locate calls result in normal Locate calls.</p>
Normal after Quick	<p>The number of normal Locate calls that have been preceded by a quick Locate call.</p> <p>For an explanation of normal calls, see the description of Quick Locate Calls above.</p>
Normal after Quick - successful	The number of normal Locate calls that have been successful in locating the required object in the buffer pool or the BP cache and have been preceded by a quick Locate call.
Normal after Quick - failed	The number of normal Locate calls that failed and were preceded by a quick Locate call.
Normal Locate Calls	The total number of normal Locate calls.
Normal Locate Calls - successful	The number of normal Locate calls that were successful in locating the required object in the buffer pool.
Normal Locate Calls - failed	<p>The number of normal Locate calls that failed.</p> <p>A failed normal Locate call indicates that an object has to be loaded from the database or from the BP cache.</p>
Successful from Cache	The total number of successful Locate calls of objects that resided in the BP cache. This information is counted only if the previous Locate call (Normal after Quick failed or Normal Locates failed) failed. It indicates the number of database loads saved. This means, that, without the BP cache, the object would have to be loaded from the database.
Load Calls	<p>The total number of load calls made since the buffer pool has been refreshed. The load calls are correlated with the access to the system file from which the objects are read.</p> <p>The number of system file accesses is calculated as follows:</p> <ul style="list-style-type: none"> <li>● Adabas system file: The number of Load Calls plus the number of Object Loads (see below). The total number does not include Adabas RC calls.</li> <li>● VSAM system file: The number of Load Calls.</li> </ul>

Field	Explanation
Object Loads	<p>The number of times an object was loaded from a Natural system file into the buffer pool.</p> <p>As several load calls may be necessary to load a single object, this value provides the actual number of object loads made since the most recent buffer pool refresh.</p> <p>When loading an object, the buffer pool manager uses different search algorithms: see METHOD=S and METHOD=N in Buffer Pool Search Methods, Principle of Operation, Natural Buffer Pool, in the Natural Operations for Mainframes documentation.</p>
Object Loads - finished	<p>Shows how many of the object loads finished successfully.</p> <p>An object load cannot finish if the load operation is canceled due to any of the following reasons:</p> <ul style="list-style-type: none"> <li>• A concurrent object load occurred: see "Object Loads - concurrent" below.</li> <li>• During the object load, an Adabas response code occurs.</li> <li>• During the object load, a SYSBPM delete operation is executed for this object.</li> </ul>
Object Loads - concurrent	<p>The number of object loads that have been performed simultaneously for the same object:</p> <p>Concurrent object loads occur if two or more Natural sessions that run simultaneously request the same object. While an object is being loaded by one session, other sessions request the same object and start loading it before a session has finished loading. In this case, the same object is loaded more than once.</p> <p>The first session that finishes loading the object will mark the object of the other sessions to be deleted from the buffer pool. The other sessions will then stop loading the object, remove the object marked for deletion from the buffer pool and use the object loaded successfully by the first session.</p> <p>The numbers of objects calculated by the counters "Object Loads - finished" and "Object Loads - concurrent" are usually identical. The numbers only differ if the concurrent load is only detected after both sessions have finished the load.</p>
Number Loads into BP	<p>The number of times a load into the buffer pool was performed successfully.</p> <p>The load into the buffer pool (storage allocation request) can be triggered either by a load from the database or by a load from the BP cache.</p>
Number Loads BP 2nd	<p>This field is displayed if METHOD=S (selection process) is used as search method for allocating storage.</p> <p>Alloc Requests 2nd shows the number of times a storage allocation request satisfied the search criteria of Algorithm 2 as described in METHOD=S in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.</p>
Number Loads into BP - Number Load Cycles	<p>This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.</p> <p>This field indicates the number of times a search has been performed starting from the top of the buffer pool. This number gives an estimate of the frequency of cycling through the buffer pool in a wrap-around fashion.</p>

Field	Explanation
Number Loads into BP- Number Lock Retries	This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.  This field indicates the number of times a chain of locked buffer pool entries had to be unlocked, because they could not satisfy the allocation request.
Number Loads into BP- Last Cycle Start	This field is displayed if METHOD=N (next available) is used as search method for allocating storage as described in METHOD=N in Buffer Pool Search Methods in the Natural Operations for Mainframes documentation.  The time and date when Number Load Cycles was last increased.
Largest Alloc (TR)	The largest single allocation size so far requested, specified in number of text records.
Number Load Failure	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.
Number Load Failure - Sizes failing last	The number of text records that would have been required by the three most recent storage allocation requests that failed.
Locates/Loads	These statistics are expressed as a ratio using Total Locate Calls Successful and Object 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 primary indicator of performance from one buffer pool session to the next.

For details on the search methods used for allocating space in the buffer pool, see Buffer Pool Search Methods under Natural Buffer Pool Principle of Operation in the Natural Operations for Mainframes documentation.

## Buffer Pool Fragmentation

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

### To invoke Buffer Pool Fragmentation

- On the Buffer Pool Statistics main menu, enter Function Code **F**.  
Or, in the command line, enter DISPLAY FRAGMENTATION.

The Buffer Pool Fragmentation screen is displayed.

Some of the fields provided on the Buffer Pool Fragmentation are identical to the items explained in General Buffer Pool Statistics above:

- Buffer Pool Size
- Buffer Pool Address
- Text Record Section
- Text Record Size
- Number of Text Records  
(same as Text Records - Total)

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

```

      1-----10-----20-----30-----40-----50
005F0480  . . . +***      ++      . . . * . +** . ++      * . . +**+      +++++XX

```

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.
blank character	An unused text record.
XX	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 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 made to the Natural buffer pool manager.

### To invoke Internal Function Usage

- On the Buffer Pool Statistics main menu, enter Function Code **I**.  
Or, in the command line, enter DISPLAY FUNCTION.

The Internal Function Usage screen is displayed.

The statistics displayed on the Internal Function Usage 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 number and as percentage. In addition, these numbers are represented in a horizontal bar chart.

## Buffer Pool Hash Table Statistics

This function only applies to buffer pools of the type Natural.

Buffer Pool Hash Table Statistics displays statistics about hash table slots and collisions per slot. The statistics determine the efficiency of the hash algorithm used. For further information on hash tables, refer to Buffer Pool Hash Table in Natural Buffer Pool in the Natural Operations for Mainframes documentation.

The statistics are primarily intended for internal use by Software AG personnel only.

### To invoke Buffer Pool Hash Table Statistics

- On the Buffer Pool Statistics main menu, enter Function Code **H**.  
Or, in the command line, enter DISPLAY HASH.

The Hash Table Collisions screen is displayed.

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

Field	Explanation																				
Total Number of Slots	The total number of hash table slots; that is, the total possible entries in the hash table that link the object names with the location of the objects. The number of slots, that is, the size of the hash table will be calculated internally depending on the number of text records.																				
Number of Slots used	The number of slots in the hash table that have at least one object name mapped to them.																				
Number of Slots free	The number of slots in the hash table that have no object name mapped to them.																				
Max. Collisions per Slot	The maximum number of collisions of any slot. The maximum number of collisions is the longest possible search path for an object.  A collision is caused if the name of two different objects is mapped to the same slot by the hash algorithm. In this case, a collision resolution is used in order to find another slot.																				
Collisions	The number of current collisions. Depending on the collisions that occur, the table contains up to 10 rows: <table style="margin-left: 40px; border: none;"> <tr><td>0</td><td>No collision.</td></tr> <tr><td>1</td><td>1 collision.</td></tr> <tr><td>2</td><td>2 collisions.</td></tr> <tr><td>3</td><td>3 collisions.</td></tr> <tr><td>4</td><td>4 collisions.</td></tr> <tr><td>5</td><td>5 collisions.</td></tr> <tr><td>6 - 10</td><td>Between 6 and 10 collisions.</td></tr> <tr><td>11 - 15</td><td>Between 11 and 15 collisions.</td></tr> <tr><td>16 - 20</td><td>Between 16 and 20 collisions.</td></tr> <tr><td>21</td><td>More than 21 collisions.</td></tr> </table> <p>No collision means that only one object name is mapped per slot. To locate this object, you need to access the hash table once only.</p> <p>If the number of collisions is greater than zero (0), for example, <math>x</math>, <math>x+1</math> object names are mapped to the same slot. To locate one of these objects, you need to access the hash table up to <math>x+1</math>.</p>	0	No collision.	1	1 collision.	2	2 collisions.	3	3 collisions.	4	4 collisions.	5	5 collisions.	6 - 10	Between 6 and 10 collisions.	11 - 15	Between 11 and 15 collisions.	16 - 20	Between 16 and 20 collisions.	21	More than 21 collisions.
0	No collision.																				
1	1 collision.																				
2	2 collisions.																				
3	3 collisions.																				
4	4 collisions.																				
5	5 collisions.																				
6 - 10	Between 6 and 10 collisions.																				
11 - 15	Between 11 and 15 collisions.																				
16 - 20	Between 16 and 20 collisions.																				
21	More than 21 collisions.																				
Number of Slots	The number of slots related to the number of collisions.  In addition, the percentage of these slots related to all slots used is displayed.																				
Number of Slots Totaled	The same values as Number of Slots, but the values are totaled.																				

**Example of Hash Table Statistics**

```

14:36:26          ***** NATURAL SYSBPM UTILITY *****          2003-08-13
BPNAME NATGBP    - Buffer Pool Hash Table Statistics -          Type Global Nat
BPPROP OFF                                             Loc DAEF QA41

Total Number of Slots ..          523
Number of Slots used ..          475 ( 90.8 %)          Max. Collisions
Number of Slots free ..          48 ( 9.1 %)          per Slot ..... 7

Collisions          Number of Slots          Number of Slots Totalled
  0                   0 ( 0.0 %)                   0 ( 0.0 %)
  1                  164 ( 34.5 %)                   164 ( 34.5 %)
  2                  194 ( 40.8 %)                   358 ( 75.3 %)
  3                   96 ( 20.2 %)                   454 ( 95.5 %)
  4                   16 ( 3.3 %)                   470 ( 98.9 %)
  5                    4 ( 0.8 %)                   474 ( 99.7 %)
  6 - 10              1 ( 0.2 %)                   475 ( 100.0 %)

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          Exit Last          Flip          Canc
    
```