

CONTENTS

List of Figures.....	v
1 Improving Domino and DB2 performance	1
1.1 Setting the size of the buffer pool to improve performance	1
2 Controlling the NSF data in DB2	3
2.1 Reorganizing the data in a table	4
2.1.1 Running the command nCompact -B to reorganize tables and indexes associated with Notes databases	5
2.1.2 Running DB2DART/LHWM to obtain reorganization options	5
2.2 Running statistics to optimize performance	6
3 Modifying the database sort heap and lock list parameter values	7
3.1 Sort heap configuration parameter	8
3.1.1 Troubleshooting the sort heap configuration parameter value	8
3.2 Lock list configuration parameter	8
4 Using the autoconfigure command to set database configuration parameters ...	10
5 Using indexing to improve DB2 performance	12
5.1 Using an index to automatically order columns	13
6 Using SQL Assist to create an SQL statement.....	15
7 Generating a database monitor snapshot	15
7.1 Saving dynamic SQL snapshot results in a file.....	16

7.2 Using a snapshot to detect negative locking behavior..... 18

LIST OF FIGURES

Figure 1 Alter Buffer Pool dialog box	3
Figure 2 Reorganize Table dialog box.....	4
Figure 3 Run Statistics dialog box	7
Figure 4 Database Configuration dialog box	9
Figure 5 Create Index dialog box.....	14

1 Improving Domino and DB2 performance

This document describes procedures for optimizing the performance of an IBM® DB2® server running IBM® Lotus® Domino® 7. The information applies to configurations that utilize medium-size servers because the parameters are tuned for that type of configuration. Other hardware configurations may require additional adjustments.

Note the following additional configuration requirements:

- If your configuration includes a DB2 Access Server, use DB2 Access Server 8.2 with IBM Lotus Domino 7.
- If your configuration includes IBM® AIX®, use IBM AIX for Domino 7.

If you are not using the appropriate software versions, you will need to upgrade. For information about upgrading software, see Lotus Domino 7 Administrator Help at <http://www-10.lotus.com/ldd/notesua.nsf>.

Complete as many of these procedures as necessary to improve the performance of your DB2 server.

- Set the size of the buffer pool
- Control the NSF data in DB2
- Modify the database sort heap and lock list performance
- Use the autoconfigure command to automatically set database configuration parameters
- Use indexing to improve DB2 performance
- Use SQL Assist to create an SQL statement
- Generate a database monitor snapshot

Other recommended documentation and resources

- IBM® DB2® UDB Performance Tuning and Monitoring Workshop, Course code CF411, IBM Learning Services, Worldwide Certified Material at <http://w3-3.ibm.com/education/CourseDescriptionServlet.wss?city=&state=&countrycode=&coursecode=CF412&courseDescrLanguageId=1&actTp=-1&isfrompopup=&fdate=&ldate=>
- IBM DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.u.db.doc/core/db2idxA.htm>

1.1 Setting the size of the buffer pool to improve performance

Buffer pools are allocated as a number of shared RAM segments on the database server. Database records are read and updated in the buffer pool area of memory. If the buffer pool is large enough to keep required data in memory, less disk activity occurs. If the buffer

pool is not large enough to keep required data in memory, overall database performance can be severely affected. If performance is poor, the first DB2 parameter to check is buffer pool size.

In a Lotus Domino 7 and DB2 configuration, the buffer pool can become quite large. To improve performance, you can increase the amount of memory allotted to the buffer pool for the initial DB2 database that was created when you installed Lotus Domino and DB2. By default, the DB2 database that is automatically created is named DOMINO, unless you specified a different database name. The buffer pool has the same name as the initial database.

To determine how much memory can be allotted for the buffer pool, subtract the amount of memory consumed from the total amount of memory. Assign the difference to the buffer pool. Buffer pool sizes are specified in numbers of pages, which are fixed blocks of data read from and written to disk. For each buffer pool, the size consumed is the page size (in kilobytes) times the buffer pool size. Figure 1 shows sample buffer pool values.

Note: To determine the amount of memory that is consumed, review the information in the DB2 Control Center or use SQL to query the SYSCAT buffer pool table.

Use the DB2 Control Center to locate the Buffer Pool configuration parameter for the first DB2 database that you created.

1. Open the DB2 Control Center.
2. Click All Databases.
3. Locate the initial DB2 database you created, and then open the database. The default database name is DOMINO.
4. Locate and open the buffer pools folder.
5. Select the buffer pool, right-click, and then click Alter.

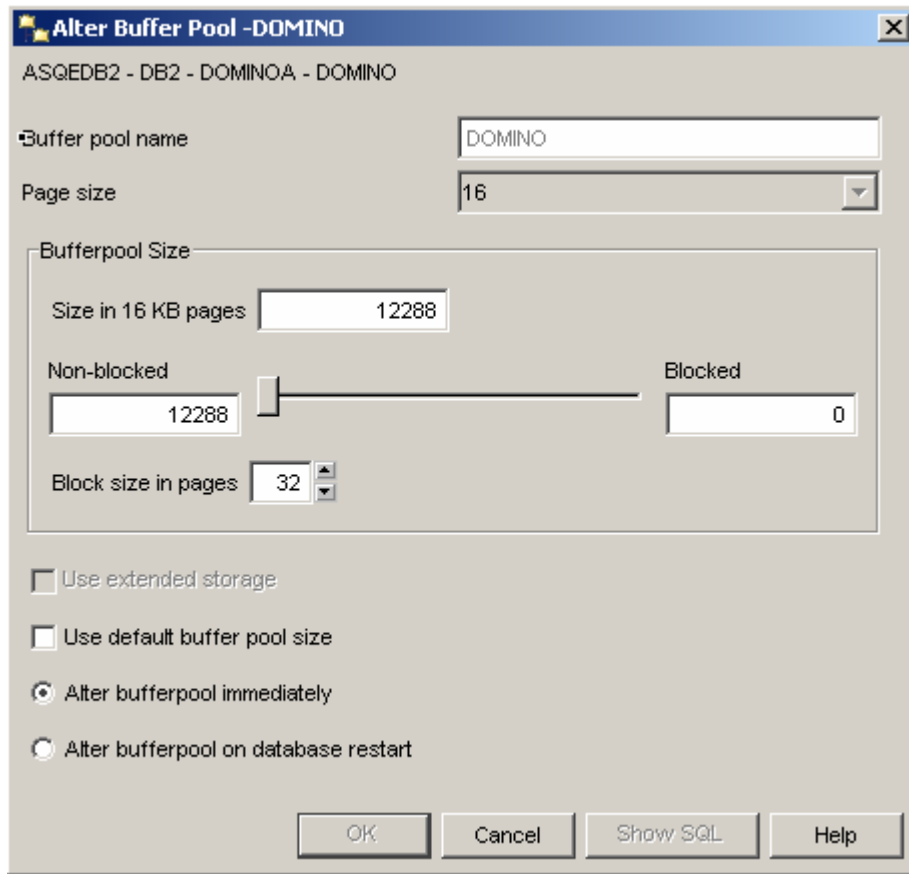


Figure 1 Alter Buffer Pool dialog box

6. Assign the largest amount of memory to the DOMINO buffer pool. Assign smaller amounts of memory, such as 12 KB to 16 KB, to each of the other buffer pools.
7. Click OK.

For detailed information about modifying configuration parameters in DB2, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>.

2 Controlling the NSF data in DB2

Use one of these options to control the NSF data in DB2:

- Reorganizing the data in a table
- Using the command `nCompact -B`
- Running DB2DART/LHWM

You can fine-tune the NSF data in DB2 by reorganizing the data stored in tables in your NSF database. You can also run statistics to gather information about a database table so

that the DB2 optimizer can make good decisions. The DB2 optimizer generates the access plan for resolving SQL statements.

Always coordinate with the database administrator to schedule when to reorganize the data in a table. Data reorganizations are not typically done during business hours.

2.1 Reorganizing the data in a table

When you reorganize the data in a table, the data is defragmented.

1. Open the DB2 Control Center.
2. Click All Databases, and then locate the initial DB2 database you created. The default database name is DOMINO.
3. Open the database, and then click Tables.
4. Select the table that you want to reorganize, and then right-click.
5. Click Reorganize to display the Reorganize Table dialog box.

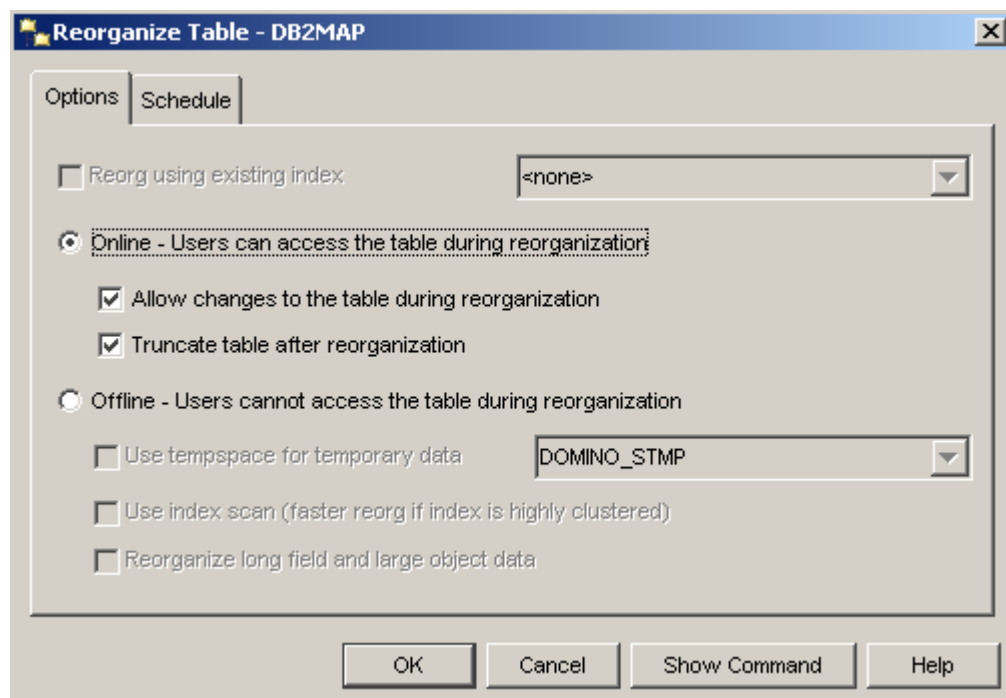


Figure 2 Reorganize Table dialog box

6. For optimum performance, select these settings:
 - a) Online – Users can access the table during reorganization
 - b) Allow changes to the table during reorganization
 - c) Truncate table after reorganization – The table size is reduced to decrease the overhead of reserving and deleting storage for temporary tables.

For information about the fields on the Reorganize Tables dialog box, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>.

2.1.1 Running the command nCompact -B to reorganize tables and indexes associated with Notes databases

The Domino add-in task, nCompact -B, performs basic in-place DB2 reorganizations on all tables and indexes associated with the DB2 enabled Notes database that you specify. The reorganization operation is performed on the entire set of tables that can be shared by multiple Notes databases that are enabled for DB2. Running nCompact -B against a directory does not perform duplicate or redundant reorganizations; that is, DB2 table and index reorganizations occur only once per table. If Domino is not running, a more aggressive offline reorganization is performed. During this reorganization, an attempt is made to move long data fields and large binary objects (LOBs) while using a temporary system-managed tablespace as a scratch pad for temporary data. This allows for maximum efficiency.

To run the nCompact -B command, from the Domino command prompt type

```
nCompact -B
```

Note: Run nCompact -B with an uppercase B, not a lowercase b.

2.1.2 Running DB2DART/LHWM to obtain reorganization options

To determine the best method for consolidating your data so that the high water mark (HWM) can be lowered, run DB2DART/LHWM on the tablespace containing the DB2 enabled Notes database. Lowering the HWM allows you to reduce the size of the tablespace.

Note: If you are using the DB2 Control Center, but the DB2 database is not local, create a local alias for the database before connecting.

Complete these steps to run the command to list tablespaces, as well as other DB2 commands similar to this one.

1. Before entering the command, connect to the first DB2 database that was created. For example, if that DB2 database is named DOMINO, enter one of these commands according to whether you need to enter a user ID and password:
 - o db2 connect to database DOMINO
 - o db2 connect to database DOMINO user UserID
 - o db2 connect to database DOMINO user UserID using Password
2. To generate a list of tablespace IDs required for DB2DART, from the DB2 Command Line Processor (CLP) run the following command:

```
db2 list tablespaces show details
```

The output from the command, db2 list tablespace show details indicates the number of free pages that exist in the tablespace, as well as the HWM. The desired HWM

is the value derived by subtracting totalPages from freePages. Use the resulting value as the target HWM when running DB2DART/LHWM.

2.2 Running statistics to optimize performance

To optimize performance, run statistics daily on your most volatile databases, and run statistics on your largest databases on weekends. You can then use the information in the statistics to help you determine how to optimize performance.

1. Open the DB2 Control Center.
2. Click All Databases, and then locate the initial DB2 database you created. The default database name is DOMINO.
3. Open the database, and then click Tables.
4. Select one or more tables on which you want to run statistics.
5. Right-click, and then choose Run Statistics. The Run Statistics dialog box appears.

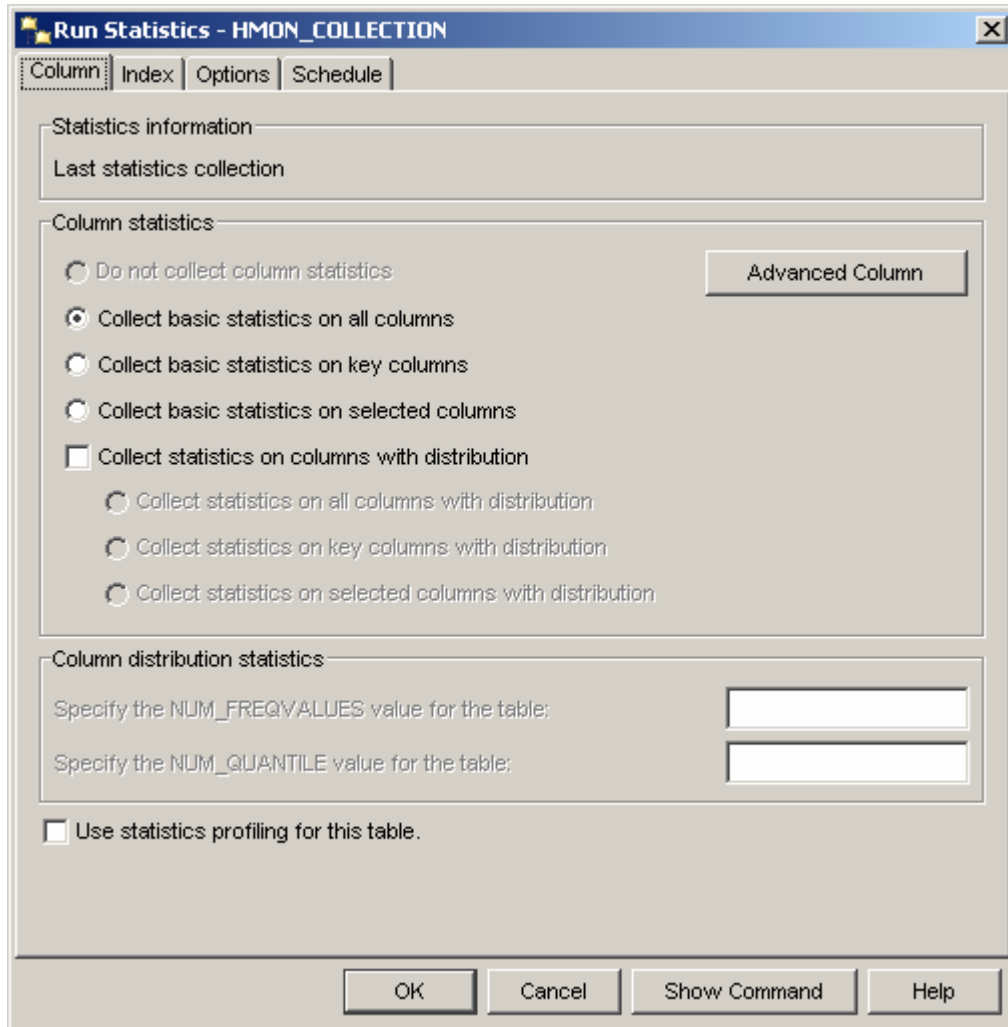


Figure 3 Run Statistics dialog box

For information about the fields on the Run Statistics dialog box, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>

6. To schedule when statistics are run, click the Schedule tab, and then choose Enable Scheduler.

3 Modifying the database sort heap and lock list parameter values

You can improve DB2 performance by increasing or decreasing the amount of memory allocated to the sort heap and lock list database configuration parameters.

3.1 Sort heap configuration parameter

The sort heap is the area where data is sorted. Sort heap is the maximum number of private memory pages used for private sorts, or the maximum number of shared memory pages used for shared sorts. For private sorts, sort heap affects agent private memory. For shared sorts, sort heap affects the database shared memory. Each sort has a separate sort heap that is allocated as needed by the database manager. If directed by the DB2 optimizer, a smaller sort heap than the one specified by this parameter is allocated.

3.1.1 Troubleshooting the sort heap configuration parameter value

The sort heap value is too small if you see this DB2 command line Interface message:

```
SQL0955C Sort memory cannot be allocated to process the statement.  
Reason code = reason-code.
```

The sort heap value is too large if you see this DB2 command line Interface message:

```
SQL5155W The update completed successfully. The current value of  
SORTHEAP may adversely affect performance.
```

To check the sort heap value, enter this command at the DB2 command line interface:

```
db2 get db cfg for <dbname>
```

To check the value of sheapthres, enter this command at the DB2 command line interface:

```
db2 get dbm cfg
```

If you see several occurrences of the following message, a sort overflow is occurring with your application. Review the access plan and if the sort is not required, remove the sort.

```
sort statement overflows
```

3.2 Lock list configuration parameter

The lock list database configuration parameter designates the amount of storage that is allocated to the lock list. There is one lock list per database and it contains the locks held by all applications concurrently connected to the database. Locking is the mechanism that the database manager uses to control concurrent access to data in the database by multiple applications. Both rows and tables can be locked. The database manager can also acquire locks for internal use.

If the lock list parameter is too small, DB2 may consolidate multiple row locks into an exclusive table lock. If the DB2 monitor snapshots show numerous row locks consolidated to table locks, consider increasing the size of the lock list parameter. Negative locking behavior has a severe impact on performance.

For information about using a DB2 monitor snapshot to detect negative locking behavior, see the topic “7.2. Using a snapshot to detect negative locking behavior.”

When the Lotus Domino server that is enabled for DB2 creates and configures the DB2 database, lock list is set to 2,500 (at 4 KB pages this is 10 MB of space.) For very active servers, you may need to increase this value to 5,000 or even 10,000. You can also run the autoconfigure command to obtain recommendations.

For information about the autoconfigure command, see the topic “4. Using the autoconfigure command to set database configuration parameters.”

You can increase the lock list database configuration parameter online but you cannot decrease it online. To decrease the lock list value, you must reactivate the database.

1. Open the DB2 Control Center.
2. Click All Databases, and then locate the initial DB2 database you created. The default database name is DOMINO.
3. Select the database name, and then right-click.
4. Choose Configure Parameters.

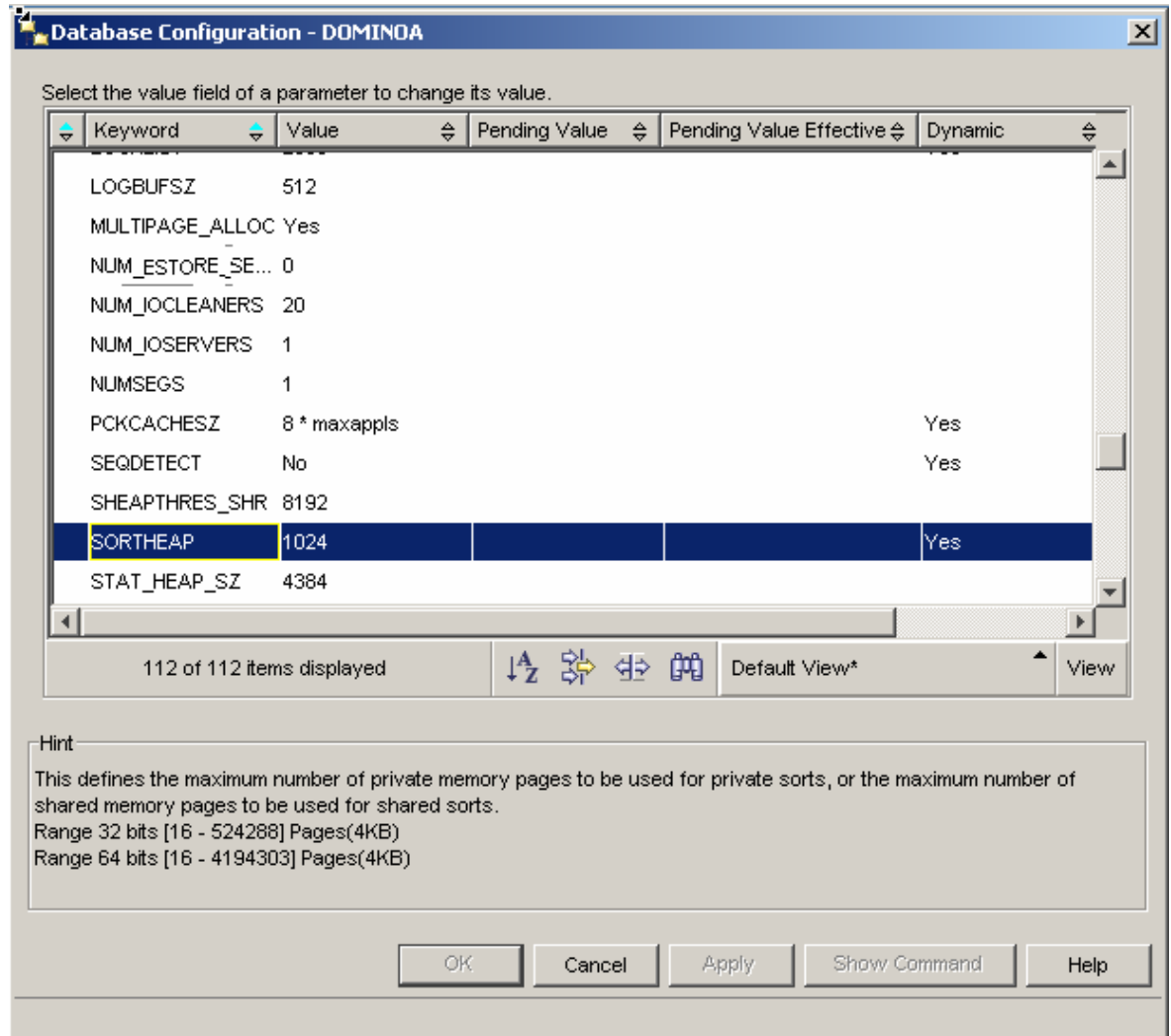


Figure 4 Database Configuration dialog box

5. In the Performance section of the Database Configuration dialog box, select the parameter keyword SORTHEAP. In the value field, enter a greater value than the value that is displayed.
6. Select the parameter keyword LOCKLIST, and specify a value that is two times greater than the value that is displayed.

For information about the parameters on the Database Configuration dialog box, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>.

4 Using the autoconfigure command to set database configuration parameters

Run the autoconfigure command after the Lotus Domino databases have been created and data has been added (is_populated yes).

The autoconfigure command recommends optimal database settings based on the available resources, internal statistics gathered by DB2, and input parameters that provide information about the Lotus Domino server requirements. Run the autoconfigure command to improve system performance.

The first time you run the autoconfigure command, include the Apply parameter as well as the parameters shown in the table, and set the parameter Apply to None to view the autoconfigure command recommendations. The following example shows the autoconfigure command with the Apply parameter:

```
db2 connect to your_database
db2 autoconfigure using num_local_apps 500
num_remote_apps 500
isolation CS
mem_percent 60
workload_type simple
num_stmts 40000
tpm 150000
is_populated yes
admin_priority performance
bp_resizeable yes
Apply None
```

After that initial test of the autoconfigure command with Apply set to None, rerun the command with Apply set to db and dbm, as shown in the following example:

```
db2 connect to your_database
db2 autoconfigure using num_local_apps 500
num_remote_apps 500
isolation CS
mem_percent 60
workload_type simple
num_stmts 40000
tpm 150000
is_populated yes
admin_priority performance
bp_resizeable yes
```

Apply db and dbm

Run the autoconfigure command with the parameters shown in the following table.

Parameter Keyword	Valid values	Default value	Recommended setting	Explanation
admin_priority	Performance Recovery Both	Both	Performance	Optimize for better performance (more transactions per minute) or better recovery time.
bp_resizeable	Yes No	Yes	Yes	Whether buffer pools are resizeable.
is_populated	Yes No	Yes	Yes	Whether the database is populated with data.
isolation	RR RS CS UR	RR	CS	Isolation level of applications connecting to this database. RR – Repeatable Read RS – Read Stability CS – Cursor Stability UR – Uncommitted Read
mem_percent	1–100	80	60	Percentage of memory to dedicate. If other applications (excluding the operating system) are running on this server, set mem_percent to less than 100.
num_local_apps	0 – 5,000	0	500	Number of connected local applications.
num_remote_apps	0 – 5,000	10	500	Number of connected remote applications.
num_stmts	1 – 1,000,000	10	40,000	Number of statements per unit of work.
tpm	1 – 200,000	60	150,000	Transactions per minute.
workload_type	Simple Mixed Complex	Mixed	Simple	Simple workloads tend to be I/O intensive and mostly transactions, whereas complex workloads tend to be CPU intensive and mostly queries.

If the computer is running only DB2, increase the parameter mem_percent. If other applications are running on the same computer with DB2, decrease mem_percent.

Note: The LOG* db cfg parameters only apply if you do not configure archive-style transaction logging, that is, you must set LOGRETAIN=No.

Use the autoconfigure parameters to allow the autoconfigure command to define values for multiple configuration parameters and to determine the scope of the application of those parameters. The scope can be set to one of these values:

Scope	Explanation
NONE	None of the values are applied
DB ONLY	Only database configuration and buffer pool values are applied
DB AND DBM	All parameters and their values are applied

If you choose to apply the changes that the autoconfigure command recommends for your database and database server, explicitly reset the following parameters. The autoconfigure command choices for these parameters are inadequate for Lotus Domino requirements. From the DB2 Command Line Processor, use the following commands to update the configuration parameters to the indicated values.

```
db2 update db cfg for YourDB2database using LOGFILSIZ 8192
db2 update db cfg for YourDB2database using LOGPRIMARY 20
db2 update db cfg for YourDB2database using LOGSECOND 4
db2 update db cfg for YourDB2database using PCKCACHESZ 50000
```

For more information about using the autoconfigure command, see the DB2 Information Center at

<http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>.

5 Using indexing to improve DB2 performance

An index is a set of one or more keys, each pointing to rows in a table. An index provides efficient access to rows in a table by creating a direct path to the data through pointers. An index optimizes data retrieval without performing a lengthy sequential search.

Indexes optimize two operations in DB2: table scans and sorts. To understand how indexes optimize table scans, assume you have this SQL statement:

```
SELECT #noteid, Subject FROM MyDAV.DAV1 WHERE From = 'Joe' AND
DATE(#modified) > '2005-01-01'
```

To resolve the query, data is traversed to satisfy the WHERE clause. Without an index, performance degrades linearly as data increases in MyDAV.DAV1. An index that only contains From will cause performance to degrade to a lesser degree, but the #modified column in the WHERE clause still forces the rows to be read. There are two possible indexing approaches to resolve this problem.

- Option One -- Using an index with From, #modified does not access rows until the #noteid and Subject are fetched for the result data.
- Option Two -- Using two indexes, one with From and one with #modified, also avoids accessing rows. The indexes also optimize other queries where the column From or #modified is used in the WHERE clause without the other column.

To understand how indexes optimize sorts, assume you have this SQL statement:

```
SELECT #noteid, Subject FROM MyDAV.DAV1 WHERE From = 'Joe' AND
DATE(#modified) > '2005-01-01' ORDER BY From, #modified
```

A sort will occur unless you use indexing Option One (shown above).

The following SQL statement generates a sort unless you have an index without all four columns in it, in the order in which the columns appear in the clause ORDER BY.

```
SELECT #noteid, Subject FROM MyDAV.DAV1 WHERE From = 'Joe' AND
DATE(#modified) > '2005-01-01' ORDER BY #noteid, Subject, From,
#modified
```

Not all sorting is expensive. If the WHERE clause is sufficiently selective, for example, returning a few hundred rows, sorts are tolerable and the index overhead is unnecessary. Sorting costs need to be analyzed through tools such as SQL snapshots.

For more information about SQL DB2 database monitor snapshots, see the topic “7 Generating a database monitor snapshot.”

In general, you may want to use an index if you generate an SQL DB2 database monitor snapshot and the snapshot data shows that you have a slow-running query and one or both of the following values is high:

- Buffer pool data logical reads
- Buffer pool data physical reads

You can use the dynamic SQL snapshot to determine which SQL statements require the most time to execute. Use the DB2 Command Editor to verify whether those statements are using the correct access plan, that is, to determine whether the SQL statement uses a DB2 index versus a table scan.

5.1 Using an index to automatically order columns

Use indexing to automatically order columns. In an index, column-name identifies a column that is to be part of the index key.

1. Open the DB2 Control Center.
2. Click All Databases, and then locate the initial DB2 database you created. The default database name is DOMINO.
3. Open the database, and then click Tables.
4. Select the table whose columns you are indexing, and then right-click.
5. Choose Create Index.

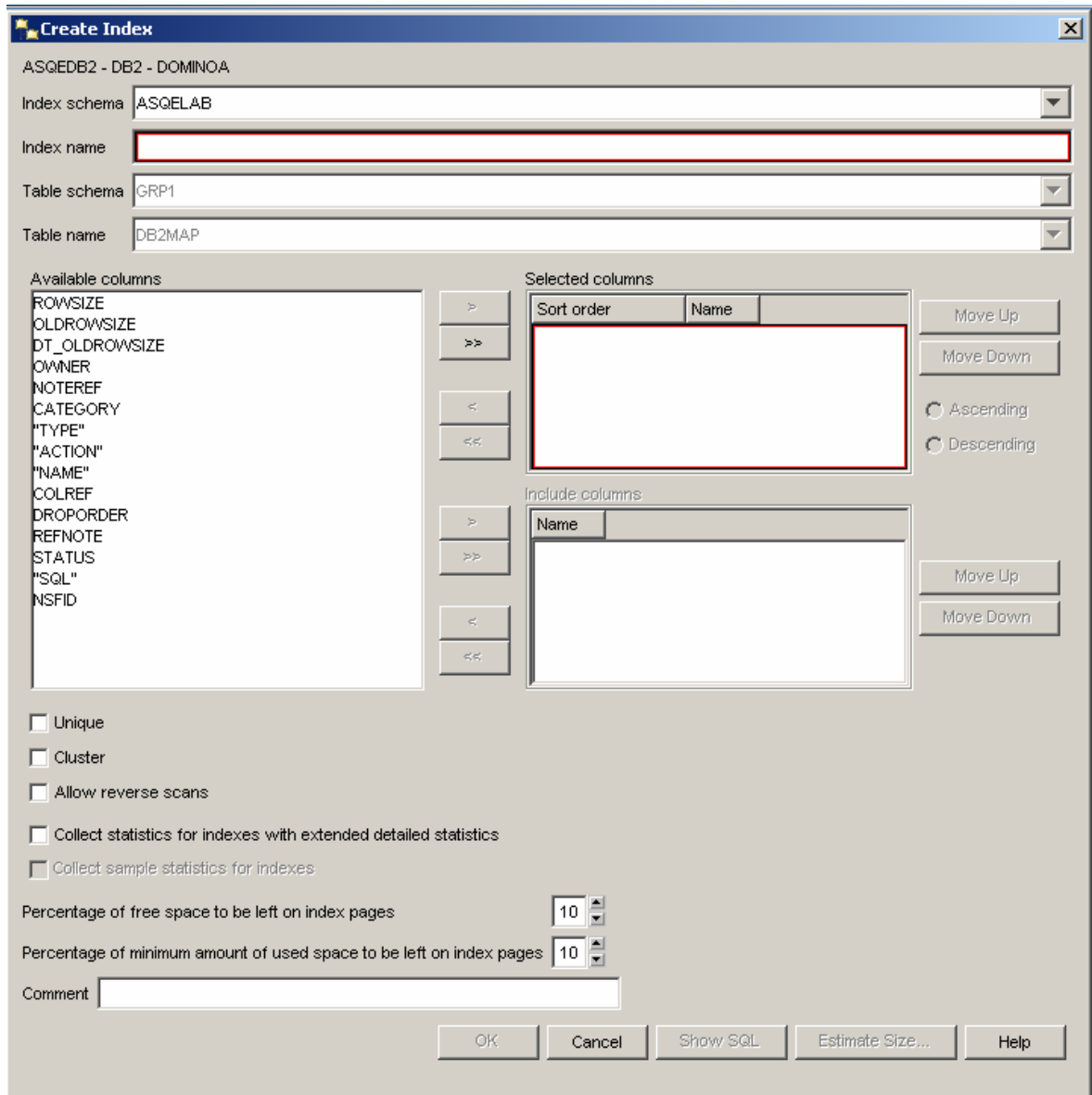


Figure 5 Create Index dialog box

6. Select the columns for which you want to establish a new sort order, and then click the > push button to move the selected column or columns to the Selected columns list box. The order in which the columns appear in the Selected columns list box is the order in which they are stored in the index.
7. (Optional) The Include columns list box is activated when you select the Unique check box. Use the Include columns list box to select additional columns to be included in the index, but not as part of the unique index key.

For information about the fields on the Create Index dialog box, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>.

8. Click OK.

6 Using SQL Assist to create an SQL statement

Use SQL Assist to create an SQL statement and to build or modify SQL SELECT, INSERT, UPDATE, or DELETE statements. Use the Access Plan feature to obtain a graphical representation of how the query resolves.

1. Open the DB2 Command Editor.
2. Click Add.
3. Specify the target database from the displayed list.
4. If you are prompted, enter a user name and password.
5. Use SQL Assist to create the SQL statement.

For information about the SQL Assist features, see the DB2 Information Center at <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/core/db2idxA.htm>

6. (Optional) Click Access Plan to generate a graphical representation of how the SQL statement will resolve.

7 Generating a database monitor snapshot

A snapshot is a point-in-time view of the monitoring elements being analyzed, some of which are cumulative. Cumulative elements contain information relative to the start of monitoring and ending at the time of the snapshot. Other (noncumulative) elements vary according to the activity on your system.

DB2 contains several default monitoring switches. Each of the monitoring switches gathers data for a specific configuration parameter. DB2 contains these default monitoring switches:

Configuration parameter	Default database monitor switch
Buffer Pool	DFT_MON_BUFPOOL = ON
Lock	DFT_MON_LOCK = OFF
Sort	DFT_MON_SORT = OFF
Statement	DFT_MON_STMT = OFF
Table	DFT_MON_TABLE = OFF
Unit of work	DFT_MON_UOW = OFF

You can take a snapshot of the data that is gathered during the monitoring session for any configuration parameter whose switch is set to On. To generate a snapshot for diagnostic purposes, the DB2 monitors must be turned on.

Note: By default, the timestamp parameter is on, but it is best to explicitly set timestamp to on because it is possible to turn timestamp off. This is the timestamp

of the log being processed. Timestamp is a DB2 monitor switch setting that you can turn on or off just as you would any monitor switch.

1. Generate a list of default monitor switches by entering this command from the DB2 CLP:

```
DB2 GET MONITOR SWITCHES
```

2. Enable monitors by using the same process that you will later use to obtain the snapshot. On Microsoft Windows platforms, use a cmd.exe session, while on AIX/Linux/Unix, use a shell.

3. Generate a database monitor snapshot by entering this command from the DB2 CLP:

```
GET SNAPSHOT FOR DATABASE ON <service>
```

7.1 Saving dynamic SQL snapshot results in a file

When you generate an SQL snapshot, the snapshot output is displayed on the computer monitor. It is not saved to a log file. To save the snapshot in a file so that you can review the information at a later time, use this command:

```
db2 connect to YourDatabase
      db2 update monitor switches using sort on
      lock on
      table on
      bufferpool on
      uow on
      statement on
      timestamp on
```

To obtain a snapshot and to use I/O redirection to save the output in a file, enter the following commands during the same process:

```
db2 get snapshot for dbm
      >db2mon.out
db2 get snapshot for all on YourDatabase
      >>db2mon.out
db2 list applications show detail
      >>db2mon.out
```

The greater than symbol (>) writes the output of the command to a file and overwrites existing text in the file. Two greater than symbols (>>) append the output of the command to a file instead of overwriting existing text.

A command sequence and its results are shown in the following table:

Command Sequence	Result
db2 get snapshot for dbm >db2mon.out	Puts the dbm snapshot in the file db2mon.out and overwrites anything that was in that file
db2 get snapshot for all on YourDatabase >>db2mon.out	Adds the database snapshot to db2mon.out

db2 list applications show detail >>db2mon.out	Adds information about applications to db2mon.out
--	---

The following is a sample of the output generated for one statement:

Output	Explanation
Number of executions = 20	Times the query has run
Number of compilations = 1	Times the SQL compiler and optimizer have processed the statement
Worst preparation time (ms) = 3	Worst case, compiler and optimizer processing
Best preparation time (ms) = 3	Best case, compiler and optimizer processing
Internal rows deleted = 0	Temp
Internal rows inserted = 0	Temp
Rows read = 0	Including temp
Internal rows updated = 0	Temp
Rows written = 0	Including temp
Statement sorts = 20	Times DB2 sorted data during the execution of this statement. Usually = Number of executions
Statement sort overflows = 0	Sort space spilling to disk. Not a good idea to sort
Total sort time = 0	Sec.ms -- shows the true cost of sorting across all executions. In this case, sorts are inexpensive
Buffer pool data logical reads = 0	Times a page containing data rows was accessed without reading data from disk
Buffer pool data physical reads = 0	Times a page containing data rows was accessed and data was read from disk
Buffer pool temporary data logical reads = 0	Times a page containing temporary data rows (for example, sorted data) was accessed without reading data from disk
Buffer pool temporary data physical reads = 0	Times a page containing temporary data rows (for example, sorted data) was accessed and data was read from disk
Buffer pool index logical reads = 1300	Times a page containing index data was accessed without reading data from disk
Buffer pool index physical reads = 0	Times a page containing index data was accessed and data was read from disk
Buffer pool temporary index logical reads = 0	Times a page containing temporary index data (for example, sorted data) was accessed without reading data from disk
Buffer pool temporary index physical reads = 0	Times a page containing temporary index data (for example, sorted data) was accessed and data was read from disk
Total execution time (sec.ms) = 0.104943 (across all executions)	
Total user cpu time (sec.ms) = 0.100145 (")	
Total system cpu time (sec.ms) = 0.010015 (")	
Statement text = SELECT nsfid, viewwid, collation, branch, hassubcategory, refunid, COUNT(*) FROM	The actual statement as DB2 saw it

<pre>GRP3.ND002115F385256FF0 WHERE refunid IN (?, ?) AND nsfid=? AND viewid = ? AND collation = ? AND branch = ? AND hassubcategory = ? GROUP BY viewid, nsfid, collation, branch, hassubcategory, refunid ORDER BY viewid, nsfid, collation, branch, hassubcategory, refunid</pre>	
--	--

7.2 Using a snapshot to detect negative locking behavior

Use the data that is gathered in the snapshot to detect negative locking behavior. To do so, in the snapshot locate the section titled "Database Lock Snapshot." A lock entry in your snapshot will look similar to this example:

```
Lock Name = 0x0D000700000000000000000000000054
Lock Attributes = 0x00000000
Release Flags = 0x00000001
Lock Count = 1
Hold Count = 1
Lock Object Name = 7
Object Type = Table
Tablespace Name = GRP8
Table Schema = GRP8
Table Name = NSFNOTE
Mode = X
```

The data in this snapshot indicates that the lock list setting is too small. DB2 consolidated multiple row locks into one exclusive table lock. This has severe consequences for system performance.

If DB2 database monitor snapshots show numerous row locks escalated to table locks, consider increasing the size of the lock list parameter.

For information about setting the lock list parameter, see the topic "3 Modifying the database sort heap and lock list parameter values."



© Copyright IBM Corporation 2006

IBM United States of America

Produced in the United States of America

All Rights Reserved

IBM, the IBM logo, AIX, DB2, DB2 Universal Database, Domino, and Lotus are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries or both.

Other company, product and service names may be trademarks or service marks of others.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Information in this paper as to the availability of products (including portlets) was believed accurate as of the time of publication. IBM cannot guarantee that identified products (including portlets) will continue to be made available by their suppliers.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.