IBM® Smart Analytics System

# Best practices

## Frequently Asked Questions on Database Administration for an IBM Smart Analytics System

***Alvin Law***
*Senior IT Consultant, IBM*

***Enda Mccallig***

*DB2 QA Engineer, IBM*

***Garrett Fitzsimons***

*Warehouse Best Practices Specialist, IBM*

Issued: December 2011

# Executive Summary

This paper answers some frequently asked questions that relate to the administration of the data warehouse database in IBM® Smart Analytics System environments.

The Frequently Asked Questions in this paper are organized into the following categories:

- Database implementation and design.
- Database administration and maintenance.
- Advanced statistics maintenance.
- Troubleshooting and resolution.

The questions and answers in this document apply to several generations of the IBM ®Smart Analytics System environment with configurations based on System x® and Power Systems™ servers. The IBM Smart Analytics System is an evolution of the IBM® InfoSphere Balanced Warehouse. Both are based on the same storage and database design principles. This paper uses the term IBM Smart Analytics System except when referring to specific InfoSphere Balanced Warehouse configurations. Unless otherwise indicated, all content applies to V9.5 and V9.7 of DB2® Database for Linux®, UNIX, and Windows® software.

This paper complements the existing paper titled "Frequently asked questions about system maintenance with the IBM Smart Analytics System" which is available on developerWorks.

Consult your IBM Smart Analytics System User Guide and associated documentation for further details on some of the topics discussed in this paper. When you intend on making changes to configuration settings in your IBM Smart Analytics System installation record, contact IBM support to help ensure that these changes do not have unanticipated consequences.

# 1. Database Implementation and Design

## *1.1 Can I create additional database partition groups?*

### Answer:

Avoid adding database partition groups and create them only in special circumstances after you have consulted with IBM Support.

A simple rule of thumb is to consult IBM Support when you wish to make changes to the configuration as stated in your IBM Smart Analytics System Installation Record.

### Additional Information:

6 database partition groups are created during system deployment. All database partition groups are created by default when the database is created.

2 database partition groups are created across all database partitions:

- IBMTEMPGROUP
- IBMDEFAULTGROUP

3 database partition groups are created on the administration database partition located on the first user module:

- IBMCATGROUP
- SDPG
- DWECONTROLGROUP

1 database partition group is created across all database partitions located on each data node.

- PDPG

Create table spaces in one of the existing database partition groups.

## *1.2 Can I create new file systems to hold any new table spaces I create?*

**Answer:**

No. When your system is deployed, dedicated file systems are created and configured for all table spaces. When high availability is configured, these file systems are directly linked to the failover configuration and must not be changed.

**Additional Information:**

For IBM Smart Analytics System 5600, 7600, and 7700 configurations, create all table space containers under the `/db2fs` file system. Use the existing naming convention. For the 5600, for example, all table space containers for the first database partition must be created under the `/db2fs/`*`instancename`*`/NODE0001/`*`databasename`* directory.

For the InfoSphere Balanced Warehouse E7000 and E7100, create all table space containers under the `/db2fs`*`L`*`p`*`P`* file systems, where *`P`* is the Balanced Partition Unit (BPU) number and *`L`* starts at 1 and increments up to the number of LUNs in the BPU. Again, follow the existing naming convention. For the E7000, for example, all table spaces containers must be created under the `/db2fs`*`L`*`p`*`P`*`/`*`instancename`*`/`*`databasename`* directory.

## *1.3 How should I organize my tables and indexes into table spaces?*

**Answer:**

This topic is covered in the IBM Smart Analytics System User Guide and in the best practices papers including the paper "Data life cycle management".

http://public.dhe.ibm.com/software/dw/dm/db2/bestpractices/DB2BP_Data_Life_Cycle_1009I.pdf

When designing table spaces, consider the effect on all database operations including queries, data load, data archiving, backup, restore and database maintenance tasks.

## *1.4 What event monitors are started by default and how should they be used?*

**Answer**:

Only the `DB2DETAILDEADLOCK` event monitor is started by default. History data is not configured to be flushed to disk. Where IBM DB2® Performance Expert or IBM InfoSphere Optim™ Performance Manager 4.1.1 is active, additional event monitors will be started. Use the following `SELECT` statement to query event monitor activity and status:

```
SELECT EVMONNAME FROM SYSCAT.EVENTMONITORS WHERE
EVENT_MON_STATE(EVMONNAME)=1
```

## *1.5 What other event monitors should I consider using?*

**Answer**:

Suggestions and details of how to set up and configure monitoring in your environment are contained in the IBM Smart Analytics System User Guide.

## *1.6 What is the difference between rah and db2_all when addressing several nodes or database partitions?*

**Answer:**

The `db2_all` utility issues the command specified once for each database partition defined in the `db2nodes.cfg` configuration file. The `rah` utility issues the command specified once for each physical node.

**db2_all**

If you have 4 database partitions on a data node, `db2_all` issues the command specified 4 times, once for each database partition. The `db2_all` command sets the `DB2NODE` environmental variable as appropriate for each database partition defined in the `db2nodes.cfg` configuration file.

**rah**

If you have 4 database partitions on a data node, the `rah` utility issues the command specified just once. The `DB2NODE` environment variable is not referenced.

# 2. Database administration and maintenance

This section covers general database administration questions.

## *2.1 How do I find out how much disk space the DB2 table spaces are using?*

### Answer:

Refer to the `/db2fs` file system on each user and data node to ascertain the amount of storage each table space container is using.

Query the `SYSIBMADM.MON_TBSP_UTILIZATION` view as shown in the following sample statement:

```
SELECT * FROM SYSIBMADM.MON_TBSP_UTILIZATION ORDER BY TBSP_NAME, MEMBER
```

To get the actual disk space used by a particular table space on each node, use the following statement:

```
SELECT TBSP_TOTAL_SIZE_KB, MEMBER FROM SYSIBMADM.MON_TBSP_UTILIZATION
WHERE TBSP_NAME = 'MY_TABLESPACE1' ORDER BY MEMBER
```

To get the total used by all table spaces on each node, issue the `SELECT` statement:

```
SELECT SUM(TBSP_TOTAL_SIZE_KB) AS TOTAL_KB, MEMBER FROM
SYSIBMADM.MON_TBSP_UTILIZATION GROUP BY MEMBER   ORDER BY MEMBER
```

Load the results into a table or a spreadsheet together with a timestamp to track usage over time.

For versions prior to DB2 9.7, use the `SYSIBMADM.TBSP_UTILIZATION` view and use the `DBPARTITIONNUM` column in place of the `MEMBER` column in the preceding query.

### Additional Information:

In partitioned database environments, issuing the command `LIST TABLESPACES SHOW DETAIL` only provides information about the database partition to which you are connected.

## *2.2 How can I lower the high water mark for a table space?*

**Answer**:

In some scenarios, table spaces have empty pages. You might not be able to reduce the table space size because of the high water mark (HWM) value.  The empty space in the table space can be used for new data inserted into that table space but it cannot be used by other table spaces.

If you try to reclaim unused space from table spaces that are not managed by automatic storage (AS), you might get the following error:

```
ALTER TABLESPACE <TSNAME> RESIZE (ALL 120G)

SQL20170N  There is not enough space in the table space "TSNAME" for
the specified action. Reason code = "1".  SQLSTATE=57059
1 The amount of space being removed is greater than the amount of space
above the high-water mark.
```

For an AS table space, reduce the table space size only above the HWM value but not below it. The following output would be returned:

```
ALTER TABLESPACE <TSNAME> REDUCE

SQL1348W The table space could not be reduced in size.
```

Use the `SYSIBMADM.TBSP_UTILIZATION` view as shown in the following sample statement to determine what portion of the table space is currently used and the current HWM value for each database partition:

```
SELECT TBSP_NAME AS TS_NAME , TBSP_ID AS TS_ID ,TBSP_USED_PAGES AS
      TS_USED_PG , TBSP_FREE_PAGES AS TS_FREE_PG , TBSP_TOTAL_PAGES AS
      TS_TOTAL_PG , TBSP_PAGE_TOP AS TS_HWM , TBSP_UTILIZATION_PERCENT
      AS UTILIZ_PCT FROM SYSIBMADM.TBSP_UTILIZATION WHERE
      DBPARTITIONNUM=1 AND TBSP_NAME= '<TSNAME>' ORDER BY
      TBSP_UTILIZATION_PERCENT, TBSP_NAME
```

The following sample output is from the preceding SELECT statement:

```
TS_NAME  TS_ID TS_USED_PG TS_FREE_PG TS_TOTAL_PG TS_HWM      UTILIZ_PCT
-------------------- ----- ---------- ---------- ----------- ----------
TSNAME   172   156864     499536     656416      656400           23.89
```

The TS_HWM column in the sample output illustrated indicates the HWM value. Another method of monitoring the HWM is the db2pd command, as shown in the following example where *172* is the `TABLE SPACEID`, and ***`sampleDB`*** is the name of the database. Note that the db2pd output is for the current database partition.

```
db2pd -tablespace 172 -db sampleDB
```

If your system uses DB2 v9.7 and the table space was created under DB2 v9.7, use the ALTER TABLESPACE statement with the LOWER HIGH WATER MARK option to reduce the HWM, as shown in the following example:

```
ALTER TABLESPACE <TSNAME> LOWER HIGH WATER MARK
```

Although the command returns immediately, the process runs asynchronously. To monitor the movement of the extents of data, use the SYSPROC.MON_GET_EXTENT_MOVEMENT_STATUS table function, as shown in the following sample statement:

```
SELECT TBSP_NAME, MEMBER,NUM_EXTENTS_MOVED, NUM_EXTENTS_LEFT,
TOTAL_MOVE_TIME FROM TABLE(SYSPROC.MON_GET_EXTENT_MOVEMENT_STATUS('', -
1)) WHERE TBSP_ID=172
```

To measure the progress of the command across all database partitions (members) in a partitioned environment, use the following sample command:

```
SELECT TBSP_NAME, MEMBER,NUM_EXTENTS_MOVED, NUM_EXTENTS_LEFT,
TOTAL_MOVE_TIME
FROM TABLE(SYSPROC.MON_GET_EXTENT_MOVEMENT_STATUS('', -2))
WHERE TBSP_ID=172 ORDER BY MEMBER
```

If the output from the preceding SELECT statement consists of '-1' values, it means that either no extent movement is in progress or the process has finished.

While the asynchronous extent movement is processing, there will be high I/O activity on the system. Do not start multiple table space changes in parallel as this causes the extent reorganization to take longer. Although the extent reorganization runs online, look to issue the procedure when there is little or no production workload on the system.

The table space status is 0x80000 (MOVE_IN_PROGRESS) while the move is in progress. Verify the status of the table space either by issuing the LIST APPLICATION SHOW DETAIL command or by querying the admin view, as shown in the following statement:

```
SELECT TBSP_NAME AS TBSP_NAME, TBSP_STATE
FROM SYSIBMADM.TBSP_UTILIZATION WHERE DBPARTITIONNUM=1
AND TBSP_NAME = '<TSNAME>'

TBSP_NAME TBSP_STATE
--------- -----------------------------------------------
TSNAME    MOVE_IN_PROGRESS
```

If you try to reduce the table space size before the asynchronous process finishes, you will get the following error:

```
ALTER TABLESPACE <TSNAME> REDUCE
SQL0290N Table space access is not allowed. SQLSTATE=55039
```

If this occurs, you must enter the command again after the extent reorganization finishes.

Use the LOWER HIGH WATER MARK option for the ALTER TABLESPACE statement to lower the HWM only for table spaces created with DB2 Version 9.7 and later. If you upgraded a table space from an earlier version of the DB2 software and attempt to use the ALTER TABLESPACE statement with the LOWER HIGH WATER MARK option, the following error is returned:

```
ALTER TABLESPACE <TSNAME> LOWER HIGH WATER MARK

SQL1763N  Invalid ALTER TABLESPACE statement for table space “TS_OLD”
due to reason “11”  SQLSTATE=429BC

Table spaces must have the Reclaimable Storage attribute before
you can use clauses to reduce or lower the high water mark.
```

To determine if a table space is enabled for reclaimable storage, issue the following statement:

```
SELECT VARCHAR(TBSP_NAME, 20) AS TBSP_NAME, TBSP_TYPE,
RECLAIMABLE_SPACE_ENABLED, TBSP_FREE_PAGES
FROM TABLE (MON_GET_TABLESPACE('' ,-1)) AS T

TBSP_NAME        TBSP_TYPE  RECLAIMABLE_SPACE_ENABLED TBSP_FREE_PAGES
---------------- ---------- ------------------------- -------------
SYSCATSPACE      DMS                                0             472
USERSPACE1       DMS                                0            3648
TS_OLD           DMS                                0             256
TS_NAME          DMS                                1             896
```

There is no way to enable reclaimable storage for a pre-v9.7 table space. You must instead create a new table space and copy the table data to the new table space.

## Additional Information:

Please refer to the following technote on reducing the HWM for table spaces created before v9.7:

http://www.ibm.com/support/docview.wss?uid=swg21006526

Refer to the topic “Reclaiming unused space from table spaces is supported” in the DB2 Information Center.

## *2.3 When should I compress my tables and indexes?*

**Answer:**

The IBM Smart Analytics System configurations are designed to use deep compression. These configurations all have higher processor to I/O ratios than data warehouse configurations that were developed before data compression was available. The DB2 software provides two types of compression: data and index.

The recommendation for data compression on older configurations was to compress only large tables, however, beginning with DB2 V9.7, the recommendation is to compress all tables (which also includes small tables).

For index compression, which is available only in releases of DB2 V9.7 software, the recommendation is to use compression only if it provides substantial space savings. Implement compression in your development and QA systems to determine if index compression is suitable for your environment.

## *2.4 How can I find out if the table is compressed?*

**Answer:**

When you create a table with the `COMPRESS=YES` option, the database manager only compresses the table if it saves space.

To determine if compression is being used for each table and the degree of compression achieved for the table, connect to your database and query the `SYSIBMADM.ADMINTABCOMPRESSINFO` view. This can be done by issuing the following statement:

```
SELECT * FROM SYSIBMADM.ADMINTABCOMPRESSINFO ORDER BY TABSCHEMA,TABNAME
```

## *2.5 How can I tell which indexes are used and which indexes are not used?*

**Answer:**

During database application development, a large number of indexes might be defined on tables to guarantee that each and every query performs well.

After application development has finished and a database has gone into production, the existence of too many indexes can decrease the performance of the database. A larger number of indexes results in more work for the database system when it is executing `UPDATE`, `INSERT`, and `DELETE` (UID) operations. In addition, regular maintenance activities like `RUNSTATS` and `REORG` also take considerably long to complete in cases where many indexes are present. As a result, in order to optimize database performance, it is important to determine which indexes are necessary for query execution and which indexes can be safely dropped without effecting query runtimes.

DB2 9.7 provides a way to detect unused indexes by tracking the number of times that a particular index was used by any of the queries on the database from the time that the database was started. Because the statistics are collected from the database start up time, it is a good idea to issue this report on a database that has been active for at least a week and after a peak period workload has completed.

## Using table functions to get index information

Use the MON_GET_INDEX table function to determine index usage. The following sample statement retrieves a list of indexes most used where DB2ADMIN is the schema, DEPARTMENT is the name of the table, and -2 denotes all database partitions.

```
SELECT SI.INDSCHEMA AS INDSCHEMA, SI.INDNAME AS INDNAME,
      MGI.INDEX_SCANS, MGI.INDEX_ONLY_SCANS
FROM TABLE(MON_GET_INDEX('DB2ADMIN', 'DEPARTMENT', -2)) as MGI,
      SYSCAT.INDEXES AS SI
WHERE MGI.TABSCHEMA = SI.TABSCHEMA AND MGI.TABNAME = SI.TABNAME AND
      MGI.IID = SI.IID ORDER  BY MGI.INDEX_SCANS DESC
```

The following sample statement retrieves a list of the indexes that are least used:

```
SELECT INDNAME FROM SYSCAT.INDEXES WHERE TABSCHEMA = 'DB2ADMIN' AND
      TABNAME = 'DEPARTMENT' AND INDNAME NOT IN (SELECT SI.INDNAME FROM
      TABLE(MON_GET_INDEX('DB2ADMIN', 'DEPARTMENT', -2)) as MGI,
      SYSCAT.INDEXES AS SI
      WHERE MGI.TABSCHEMA = SI.TABSCHEMA AND    MGI.TABNAME =
      SI.TABNAME AND MGI.IID = SI.IID)
```

## Using the db2pd utility to get index information

Prior to DB2 9.7, use the db2pd utility to obtain information on table spaces. Determine the table space ID and the table ID before using the db2pd utility. The following statement shows how to determine the table ID and table space ID for a table named DEPARTMENT:

```
SELECT TABLEID,TBSPACEID FROM SYSCAT.TABLES WHERE TABNAME='DEPARTMENT'
```

After you have determined the table ID and table space ID, use a command similar to the following example to print statistics for a table named SAMPLE to an output file:

```
db2pd -db sample -tcbstats all tbspaceid=5 tableid=2 -file
db2pd_tab_dept.txt
```

## Determining when an index was last used in DB2 9.7

Starting in DB2 9.7, there is a LASTUSED column in the SYSCAT.INDEXES view that reports the date when the index was last used by any DML statement to perform a scan or used to enforce referential integrity constraints. The default value is '0001-01-01'.

To find which indexes have never been used:

```
SELECT INDSCHEMA, INDNAME FROM SYSCAT.INDEXES WHERE LASTUSED = '0001-
01-01'
```

To find the 10 most recently used indexes:

```
SELECT INDSCHEMA, INDNAME, LASTUSED FROM SYSCAT.INDEXES
WHERE LASTUSED <> '0001-01-01 ' AND OWNERTYPE = 'U'
ORDER BY LASTUSED DESC FETCH FIRST 10 ROWS ONLY
```

### Additional Information:

Related topics in the DB2 Information Center:

- "MON_GET_INDEX table function – get index metrics"

Please refer to the developerWorks article on "How to identify unused indexes in DB2 for Linux, UNIX, and Windows":
http://www.ibm.com/developerworks/data/library/techarticle/dm-0910db2unusedindex/index.html

## *2.6 Can I force the DB2 optimizer to use replicated dimensional tables?*

### Answer:

If the DB2 optimizer is not using replicated dimensional tables, set the DB2 registry variable `DB2_EXTENDED_OPTIMIZATION=FORCE_REPLICATED_MQT_MERGE`.

This enforces the use of replicated tables regardless of the Optimizers costing decisions.

### Additional Information:

For more information refer to APAR IC62086.

Please refer to the DeveloperWorks article on "Improve DB2 query performance in a business intelligence environment":
http://www.ibm.com/developerworks/data/library/techarticle/dm-0703kapoor/index.html#method2

## *2.7 How does REORG operate with temporary table spaces on a solid-state device (SSD)?*

### Answer:

SSD technology is only available on the IBM Smart Analytics System 7700 and IBM Smart Analytics System 5600S models.

The REORG command, when issued in `OFFLINE` mode, needs space to store a shadow copy of the table during processing. Default DB2 behavior is to place this shadow copy of the table into the same table space that holds the original table. Specify the `USE` clause if you want to store the shadow copy in a specific system temporary table space. If a failure occurs during the `REPLACE` phase of the `REORG` operation, the database manager needs the information in both the original table space and the temporary table space to recover from the failure.

A problem can result if the following scenarios occur during `REORG` operation:

- You issue the `REORG` utility with a `USE` clause that specifies the default `temp16k` system temporary table space, which is located on SSD storage.

- An administration node or data node fails during the REPLACE phase of the REORG operation.

The problem with this scenario is that the data located on the original SSD containers is no longer accessible. Therefore, the database manager might not be able to recover from the failure.

For this reason, when you issue the REORG command with the USE clause, create a new system temporary table space in which to store the shadow copy of the table while the REORG operation completes.

Use the following sample code as a template for your environment:

```
CREATE TEMPORARY TABLESPACE db2_reorg_tbspace IN DATABASE PARTITION
GROUP ibmtempgroup PAGESIZE 16384 MANAGED BY SYSTEM USING
('db2_reorg_tbspace')BUFFERPOOL bp_16k OVERHEAD 4.0 TRANSFERRATE 0.04
```

After you create the system temporary table space, issue the REORG command with the USE clause and specify the temporary table space you created with the previous command, for example:

```
REORG TABLE <schema_name>.<table_name> USE db2_reorg_tbspace
```

After the REORG operation completes successfully, drop the second temporary table space to prevent queries from using this table space. Use the following command syntax to drop the temporary table space:

```
DROP TABLESPACE db2_reorg_tbspace
```

## 2.8 *How do utilities affect logging?*

### Answer:

How some of the more commonly used utilities affect transaction logging follows:

**REORG**

Specification of the ONLINE or INPLACE parameters requires more transaction log space than that of OFFLINE, though OFFLINE needs to take a 'shadow' copy of the object for the duration of operation. Reference the following information:

[http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0024860.html](http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.admin.perf.doc/doc/c0024860.html)

**RUNSTATS**

Using the RUNSTATS utility generates no transaction log activity.

**IMPORT**

The import utility logs each row imported and can be intensive when importing a large volume of data. The LOAD utility is recommended over the IMPORT utility when loading data.

**LOAD**

The LOAD utility generates a minimal level of transaction log activity.

**EXPORT**

The EXPORT utility generates no transaction log activity.

## *2.9 Why should I use RUNSTATS ?*

### Answer:

Statistics are details about a database table or index that the DB2 query optimizer uses to determine the optimal data access plan. Accurate statistics are essential for query optimization. If statistics are not available for a database table or index, or if the statistics do not reflect the current state of the data, the query optimizer might choose an inefficient way to access the required data, which uses system resources inefficiently and can cause queries to take longer to complete.

### Additional Information:

The RUNSTATS command can be invoked manually or automatically. Beginning with DB2 v9.1, the default installation method for the DB2 software configures the database to collect statistics automatically. However, you might want to consider scheduling RUNSTATS during a low activity period to avoid resource contention.

The following statement can be used to determine if statistics have been collected for a specific table:

```
SELECT TABSCHEMA, TABNAME, STATS_TIME, CARD FROM SYSCAT.TABLES
WHERE TABSCHEMA like '%<table_schema>%' AND TABNAME like
'%<table_name>%' ORDER by CARD
```

If the STATS_TIME value is missing (null) from the results, it indicates that there are no statistics collected for the specified table. If the STATS_TIME value exists, it indicates the time when statistics were last collected for the table. The CARD value represents the estimated number of rows in the table. If the CARD value is "-1," the statistics for the table are missing.

For more information on optimizing queries, refer to the best practices paper on developerWorks:

www.ibm.com/developerworks/data/bestpractices/smartanalytics/queryoptimization/

Related topics in the DB2 Information Center:

- "The query optimizer" – v9.5, v9.7
- "RUNSTATS command" – v9.5, v9.7
- "Catalog statistics" – v9.5, v9.7
- "auto_maint – Automatic maintenance configuration parameter" – v9.5, v9.7

## *2.10 How are statistics collected in an IBM Smart Analytics System?*

**Answer:**

The database is partitioned across multiple database partitions. Table statistics are collected on a single database partition and then extrapolated for the other database partitions. If a table exists on the database partition to which a user is connected, the database manager always collects statistics (both synchronous and asynchronous) for that table on the current database partition. If a table does not exist on the database partition, the database manager collects statistics for that table on the first database partition of the database partition group in which the table is located.

The table statistics can be inaccurate if table data is skewed. This means that some database partitions are holding more data than other database partitions. The DB2 query optimizer might not generate the best possible data access plan if the statistics are inaccurate. If the difference in the number of rows of a table that are hosted on different database partitions is high, e.g. greater than 20%, it is preferable to collect statistics on the database partition that holds the largest number of rows.

## *2.11 How often should I update the statistics for a table?*

**Answer**:

Update the statistics only if the contents of a table have been modified considerably. The definition of "modified considerably" depends on the table contents and the importance of the data, but typically you should collect statistics after 10 percent of the content in a large fact table has been modified. Collect statistics more frequently for a small table or for dimension tables that are frequently used in queries.

Update statistics for a table in the following scenarios:

- After a `LOAD` or `REPLACE` operation
- After you reorganize tables by using the `REORG` command or the `REDISTRIBUTE DATABASE PARTITION GROUP` command.
- After you compress tables by using the `REORG` command.
- After you create a new index. Create statistics only on the index if the table data already has current statistics.
- After the underlying table for a statistical value has been modified substantially so that the view returns different rows.
- Before binding applications for which the performance is critical.

## *2.12 How can I reduce the priority assigned to the RUNSTATS utility?*

**Answer:**

Use the utility impact limit (`UTIL_IMPACT_LIM`) and utility impact priority (`UTIL_IMPACT_PRIORITY`) database configuration settings to reduce the effect of executing a DB2 utility such as `RUNSTATS` or `BACKUP` on the execution of the regular database workload.

The utility impact limit setting sets the percentage (0-100) limit of the total impact all throttled utilities can have on the overall workload of the system. A percentage of 100 denotes no throttling.

The utility impact priority setting (0-100) is used to designate a utility as a throttled utility and to prioritize that utility amongst the throttled utilities. The priority is determined by the number setting, where 1 is the lowest priority and 100 is the highest. A setting of 0 means the utility is not throttled. Specify this option for each command explicitly or alter the priority of an on-going utility execution using the SET command.

The following sample command shows how to throttle the RUNSTATS command:

```
UPDATE DBM CFG USING UTIL_IMPACT_LIM 50

RUNSTATS ON TABLE <table_schema>.<tab_name> UTIL_IMPACT_PRIORITY 100
```

The following sample command shows how to use the SET command to change the priority of a currently processing utility to 90 where <n> represents the process. This then throttles the utility and leaves it processing with a lower priority than the RUNSTATS already processing with a priority of 100:

```
RUNSTATS ON TABLE <table_schema>.<tab_name2>

SET UTIL_IMPACT_PRIORITY FOR <n> TO 90
```

where *n* is the utility ID. Use the LIST UTILITIES command to determine the utility ID.
Related topics in the DB2 Information Center:

- "SET UTIL_IMPACT_PRIORITY command" – v9.5, v9.7
- "LIST UTILITIES command" – v9.5, v9.7

## *2.13 How can I speed up statistics collection?*

**Answer:**

During `RUNSTATS` command execution: If collecting statistics using all data in the table is taking too long, specify an option to scan just a sample of the data in the table. Although the statistics are not as accurate as if you had taken a full scan, using a sample can provide a good compromise between `RUNSTATS` command execution time and better statistics. The `RUNSTATS` command includes an optional parameter, the `TABLESAMPLE` option, which allows you to select the percentage of the table you want it to scan.

The following sample command collects statistics on a table using 10 percent of the data in the table:

```
RUNSTATS ON TABLE <tabschema>.<tabname>
WITH DISTRIBUTION AND INDEXES ALL TABLESAMPLE SYSTEM (10)
```

When collecting index statistics for a table, speed up the elapsed time of the statistics collection through using several options:

Specify only the columns for which you want to collect statistics by using the `ON COLUMNS` option of the `RUNSTATS` command, as shown in the following sample command:

```
RUNSTATS ON TABLE <tabschema>.<tabname> ON COLUMNS (column1, column2)
```

Specify only the key columns, as shown in the following sample command:

```
RUNSTATS ON TABLE <tabschema>.<tabname> ON KEY COLUMNS
```

Exclude XML columns if you do not need statistics for them, as shown in the following sample command.

```
RUNSTATS ON TABLE <tabschema>.<tabname> ON ALL COLUMNS EXCLUDING XML
COLUMNS
```

During `LOAD/REPLACE`: If you are replacing the full contents of a table using the `LOAD` command with the `REPLACE` option, collecting statistics during the `LOAD` operation can be faster than performing separate operations for `LOAD` and `RUNSTATS`. Create a profile for the table and then use the `STATISTICS USE PROFILE` option with the `LOAD` command.

The following sample command uses a statistics profile to collect statistics while data is loaded into a table:

```
LOAD FROM xxx.del OF DEL REPLACE INTO <tabschema>.<tabname> STATISTICS
USE PROFILE
```

Related topics in the DB2 Information Center:

- “`RUNSTATS` command” – v9.5, v9.7
- “`LOAD` command” – v9.5, v9.7

# 3. Advanced statistics maintenance

This section focuses on advanced maintenance of statistics through the RUNSTATS command.

## *3.1 How can I use a statistics profile to collect statistics?*

**Answer**:

The recommended procedure for collecting statistics in an IBM Smart Analytics System data warehouse is to create a statistics profile for your tables. Specify a statistics profile when you invoke the RUNSTATS utility. Specifying this option configures the database manager to use the same options every time you issue the RUNSTATS command.

### Additional Information:

There are several options to choose from when you collect table statistics. After you choose the optimal options for a table (based on the number of rows, table size, or importance of the data), save them to a profile. After you have set the profile for a table, issue the RUNSTATS command with the USE PROFILE option.

For example, to issue the RUNSTATS command and specify a statistics profile, use the following syntax:

```
RUNSTATS on TABLE <table_schema>.<table_name> USE PROFILE
```

To set a profile, issue the RUNSTATS command using the following syntax:

```
RUNSTATS ON TABLE table_schema.table_name SET PROFILE [ONLY]
```

To view a statistics profile, issue the following statement:

```
SELECT TABSCHEMA, TABNAME, STATISTICS_PROFILE FROM SYSCAT.TABLES WHERE
TABSCHEMA LIKE '%xxx%'
```

A statistics profile can be used with other utilities. For example, to specify a statistics profile with the LOAD command, issue the following command:

```
LOAD FROM filename OF DEL REPLACE INTO <table_schema>.<table_name>
STATISTICS USE PROFILE
```

Related topics in the DB2 Information Center:

- "RUNSTATS command" – v9.5, v9.7

## *3.2 Can I use automatic statistics profile generation?*

**Answer**:

Although automatic statistics profile generation is available beginning with DB2 v9.1, it is currently not supported in a partitioned database environment. For this reason, the **auto_stats_prof** and **auto_prof_upd** database configuration parameters should be set to OFF.

The following database configuration settings relating to statistics collection are recommended for IBM Smart Analytics System environments. All of these settings are the default settings:

```
Automatic maintenance                     (AUTO_MAINT)      = ON
   Automatic table maintenance            (AUTO_TBL_MAINT)  = ON
      Automatic runstats                  (AUTO_RUNSTATS)   = ON
         Automatic statement statistics   (AUTO_STMT_STATS) = ON
      Automatic statistics profiling      (AUTO_STATS_PROF) = OFF
         Automatic profile updates        (AUTO_PROF_UPD)   = OFF
```

## *3.3 What are the recommended IBM Smart Analytics System profile settings?*

**Answer**:

The recommended profile settings for a particular table depend on the size of a table. To determine appropriate profile settings, begin by classifying each of your tables as small, medium, or large, based on the size of the table.

To calculate the size of a table, multiply the number of rows by the average row size. If you have collected statistics for a table, determine the number of rows (`CARD)` and the average row size `(AVGROWSIZE)` from querying the `SYSCAT.TABLES` table. This document presents recommendations for each of the three table sizes.

To make sure that all tables have at least an initial, sampled set of statistics so they can be correctly classify them by size, create a file containing the command that generates the `RUNSTATS` command script for all tables, without specifying a statistics profile. Copy and paste the following command into a text file named `initial_runstats.txt` and save it. Be sure to include the semicolon at the end of the command.

```
SELECT 'RUNSTATS ON TABLE "' CONCAT RTRIM(SUBSTR( RTRIM(TABSCHEMA)
CONCAT '.' CONCAT TABNAME,1,50)) CONCAT '" TABLESAMPLE SYSTEM (1) ;'
FROM SYSCAT.TABLES WHERE TYPE = 'T' AND CARD = -1
```

Using a DB2 command line window, connect to the database and execute the script, saving the output to a file:

```
DB2 CONNECT TO <DB_NAME>

db2 -x -tf initial_runstats.txt > tablist.txt
```

Issue the `RUNSTATS` command script you just created, saving the output to confirm that no problems occurred while the `RUNSTATS` command was executing:

```
db2 -vtf tablist.txt > tablist.out
```

Based on the output from this command, classify each of your tables as small, medium-sized, or large.

**For small tables**, collect statistics with distribution on all indexes and columns. Adjust the `CARD` value as appropriate for your definition of a small table. The following sample command uses the value 100,000 as the upper limit for a small table. The `ALLOW WRITE ACCESS` option enables users to modify data in the table while the `RUNSTATS` utility is executing; the `SET PROFILE ONLY` option prevents the collection of statistics and sets the profile for a table only.

```
SELECT 'RUNSTATS ON TABLE "'CONCAT RTRIM(SUBSTR (RTRIM(TABSCHEMA)
      CONCAT '.' CONCAT TABNAME,1,50)) CONCAT '" WITH DISTRIBUTION AND
      DETAILED INDEXES ALL ALLOW WRITE ACCESS SET PROFILE ONLY; ' FROM
      SYSCAT.TABLES WHERE TYPE = 'T' AND CARD < 100000
```

Start the script using the example below, saving the output to a file:

```
db2 -x -tf small_tables.txt > small_tables.out
```

Start the `RUNSTATS` command script you just generated, saving the output to confirm that no problems occurred while the `RUNSTATS` command was executing:

```
db2 -vtf small_tables.out > small_runstats.out
```

**For medium-sized tables**, collect statistics about all existing indexes of a table and collect data distribution statistics on each of the key columns. Do not collect statistics on non-key columns. Adjust the `CARD` values as appropriate for your definition of a medium table. The following sample command uses the value 100,000 as the lower limit and the value 10,000,000 as the upper limit for a medium-sized table.

```
SELECT 'RUNSTATS ON TABLE "' CONCAT RTRIM( SUBSTR( RTRIM(TABSCHEMA)
CONCAT '.'  CONCAT TABNAME,1,50 )) CONCAT '" WITH DISTRIBUTION ON KEY
COLUMNS AND DETAILED INDEXES ALL ALLOW WRITE ACCESS SET PROFILE ONLY;'
FROM SYSCAT.TABLES WHERE TYPE= 'T' AND CARD BETWEEN 100000 AND 10000000
```

Start the script using the example below, saving the output to a file:

```
db2 -x -tf medium_tables.txt > medium_tables.out
```

Issue the `RUNSTATS` command script you just generated, saving the output to confirm that no problems occurred while the `RUNSTATS` command was executing:

```
db2 -vtf medium_tables.out > medium_runstats.out
```

**For large tables**, collect statistics for all indexes and distribution statistics on key columns only. Create a table sample using 10% of the data by specifying the `TABLESAMPLE SYSTEM(10)` parameter instead of doing a full table scan. The reason to use a table sample is to speed up the collection of the statistics. Adjust the CARD values as appropriate for your definition of a large table. The following sample command uses the value 10,000,000 as the lower limit for a large table:

```
SELECT 'RUNSTATS ON TABLE "' CONCAT RTRIM(SUBSTR (RTRIM(TABSCHEMA)
      CONCAT '.' CONCAT TABNAME,1,50)) CONCAT '" WITH DISTRIBUTION ON
      KEY COLUMNS AND INDEXES ALL ALLOW WRITE ACCESS TABLESAMPLE SYSTEM
      (10) SET PROFILE ONLY;' FROM SYSCAT.TABLES WHERE TYPE = 'T ' AND
      CARD > 10000000
```

Start the script using the example below, saving the output to a file:

```
db2 -x -tf large_tables.txt > large_tables.out
```

Issue the `RUNSTATS` command script you just generated, saving the output to confirm that no problems occurred while the `RUNSTATS` command was executing:

```
db2 vtf large_tables.out > large_runstats.out
```

For some large tables or important tables, create a personalized profile after studying the queries that use the table. Collect extended statistics for key columns, increase the `NUM_FREQVALUES` and `NUM_QUANTILES` to better reflect the column data distribution, create column groups if applicable, and if a given column is frequently used in the LIKE predicate of SQL queries, specify the `LIKE STATISTICS` option clause.

The following sample command shows the changes recommended for large and frequently accessed tables.

```
RUNSTATS ON TABLE DEALER.CAR_PARTS ON COLUMNS ( PART_NO,CAR_BRAND,
      CAR_MODEL ,PART_DESCRIPTION LIKE STATISTICS,( CAR_BRAND,
      CAR_MODEL )) WITH DISTRIBUTION ON COLUMNS ( PART_NO
      NUM_FREQVALUES 100 NUM_QUANTILES 100 ,CAR_BRAND NUM_FREQVALUES 50
      NUM_QUANTILES 50 ,CAR_MODEL NUM_FREQVALUES 50 NUM_QUANTILES 50
      ,PART_DESCRIPTION LIKE STATISTICS) AND SAMPLED DETAILED INDEXES
      ALL ALLOW WRITE ACCESS TABLESAMPLE BERNOULLI (20) SET PROFILE
      ONLY
```

After you have set the profiles for a table, modify the scripts that you used to set the profile so that they instead execute the `RUNSTATS` command.

## Additional Information:

Related topics in the DB2 Information Center:

- "Maintenance windows" – v9.5, v9.7
- "RUNSTATS command" – v9.5 , v9.7

## *3.4 What does the "LIKE STATISTICS" RUNSTATS parameter do?*

### Answer:

When a table column is often used in the `LIKE` predicate of an SQL statement, collecting statistics about the sub-elements can help improve the response time of the query. For example:

```
SELECT COLUMN1 FROM TAB1= WHERE COLUMN5 LIKE '%ZZZ%' AND COLUMN7 LIKE
'%YYY%'
```

Improve the performance of this type of query by executing the `RUNSTATS` command with the `LIKE STATISTICS` option:

```
RUNSTATS ON TABLE SCHEMA.TAB1 WITH DISTRIBUTION ON COLUMNS (COLUMN5
LIKE STATISTICS, COLUMN7 LIKE STATISTICS)
```

The `RUNSTATS` utility might take longer to complete when you specify the `LIKE STATISTICS` option. In test environments the difference was marginal.

### Additional Information:

Related topics in the DB2 Information Center:

- "Runstats statistics about sub-elements" – v9.5, v9.7

## *3.5 What are column group statistics?*

### Answer:

When a table contains two or more columns with correlated data, use column group statistics (CGS) to detect that correlation. Suppose you have a car parts table that contains one column with car brand data and another column with car model information. Collecting column group statistics on those two columns indicates to the DB2 optimizer that individual car-parts are associated with a specific car model and brand. This helps prevent the optimizer from making estimations that would be true only if every car model existed for every car manufacturer. Collecting column group statistics can therefore help the optimizer to make better estimates of how many rows a query returns when using these columns in the `WHERE` clause or in a `GROUP BY` operation.

### Additional Information:

The following sample command uses column group statistics to indicate that the car brand and car model columns are correlated:

```
RUNSTATS ON TABLE BCUAIX.CAR_SALES ON ALL COLUMNS AND COLUMNS
((CAR_BRAND, CAR_MODEL))
```

To determine if the columns in a table are correlated, one product to use is IBM ® InfoSphere Information Server, which is a data integration software platform. One of the components in this product is the IBM InfoSphere Information Analyzer data profiler. Besides performing a number of other types of analysis (in-column analysis, column group analysis and table analysis), it can also automatically discover non-obvious correlated columns within tables.

For more information, see the main web sites for IBM InfoSphere Information Server and IBM® InfoSphere Information Analyzer:

http://www.ibm.com/software/data/integration/info_server/

http://www.ibm.com/software/data/infosphere/information-analyzer/

# 4. Troubleshooting

This section focuses on troubleshooting and problem determination for the most frequently asked questions in this area.

## *4.1 How should I manage DB2 diagnostic files?*

### Answer:

With the increasing use of autonomic technologies, DB2 servers can produce large message logging files, administrative notification log files, and event log files. This is especially true in large warehouse environments with many logical and physical partitions. When a fault occurs, the database manager can produce vast amounts of diagnostic data for first failure data capture (FODC) purposes. This increase in logging activity can lead to increased file system space consumption and manageability issues.

Allowing the `db2diag.log` file to grow until it consumes all space in the file system can lead to significant problems. Simply deleting the diagnostic log files is not a viable option because DB2 support often requests historical diagnostic data.

It is recommended that you download and use the `db2back.ksh` script from the developerWorks web site. This script includes many useful options. For example, remove diagnostic and administration notification log entries that have been produced a certain number of days ago or archive the diagnostic and administration notification log files to a new location in compressed format. Terms and conditions included in the download apply and all scripts should be tested in a non-production environment before deployment in production. Download the script from the following site.

http://www.ibm.com/developerworks/data/library/techarticle/dm-0904db2messagelogs/index.html?ca=dth-grn&=dgp-my

This script is currently the recommended method for managing diagnostic files. One consideration of using this script is that you must manually download it and configure it on your servers, and then use `crontab` to schedule it to process regularly.

For development and QA clusters, set the `DIAGSIZE` database manager parameter to a non-zero value and restart the instance; a series of rotating diagnostic log files and a series of rotating administration notification log files are used. When non-zero, the `DIAGSIZE` parameter specifies the size in MB of theses log files.

These files are called the `db2diag.n.log` and `<instance>.n.nfy` files, where n is an integer; `<instance>.n.nfy` files apply only to Linux and UNIX operating systems.

The number of `db2diag.n.log` files and `<instance>.n.nfy` files cannot exceed 10 each. When the tenth file is full, the oldest file is deleted, and a new file is created.

One advantage of this method is that a single command is used to set it up. A significant disadvantage of this method, however, is that the database manager can remove important diagnostic information that was recently produced if the set size is too small. As a result, if using this method set the `DIAGSIZE` database manager parameter sufficiently high to store a minimum of a week's worth of diagnostic files. Specifying a size that can hold the diagnostic files for longer than a week is recommended.

## *4.2 What do I do if I need help with my IBM Smart Analytics System or InfoSphere Balanced Warehouse?*

**Answer**:

The recommended method is to submit a new service request by using IBM Service Request. IBM Service Request is an online problem management tool, accessible from the following web site: https://www.ibm.com/support/servicerequest

The IBM Smart Analytics System and InfoSphere Balanced Warehouse offerings include integrated software and hardware support using a single problem queue. Your service request connects you with an IBM Remote Technical Support Representative who can help you diagnose your problem. If the problem is not immediately resolved, the technical support representative can help you create a plan to resolve it. Collect the following information before contacting support. The first three items are included in the Installation Record which is part of the hand over material for an IBM Smart Analytics System.

- Your IBM Customer Number (ICN)
- You Passport Advantage® Agreement number (PPA)
- The machine type, model number, and serial number
- A clear definition of the problem

**Additional Information:**

For more information, use the following link to search for all customer support plans for IBM Smart Analytics System offerings:
http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebDocs/?Search&Query=[HTMLDocumentName=WM_CSP]+AND+%28smart%20%29

Use the telephone number to contact IBM Software Support directly. The numbers are listed for your country here: http://www.ibm.com/planetwide/

## *4.3 What tools are available to investigate database issues?*

**Answer**:

There are a number of tools available to query the state of objects in the database. IBM DB2 Performance Expert and IBM InfoSphere Optim Performance Manager are available as default on some IBM Smart Analytics System models and are the recommended tools for managing performance. In addition, the `db2top` utility is available for real-time monitoring from the operating system command line.

Use IBM InfoSphere Optim Performance Manager to monitor database performance, receive alerts when performance indicators reach warning or critical levels, isolate performance issues, identify long running queries for tuning, view disk consumption and growth trends, configure DB2 Workload Manager, and more.

Use IBM Optim Data Studio to develop queries, format queries, view explain plans, get statistics advice, develop, debug, and deploy routines, manage inter-related object changes, schedule maintenance, and view, browse, edit, unload and load data.

Those most frequently used DB2 software utilities are listed here:

- **db2pd**

  Use the `db2pd` utility to query data for bufferpool, transaction, and memory. `db2pd` is non intrusive since it does not acquire latches or use DB2 engine resources. The `db2pd` utility might retrieve data that is changing and might reflect a different state.

- **db2mtrk**

  Use the `db2mtrk` utility to query memory. The `db2mtrk` utility can affect the database on which it is used. Do not use this utility on production systems during the online day.

- **db2dart**

  Use the `db2dart` utility to query table space data. The `db2dart` utility is non-intrusive. However, it can return inconsistent data where there are active database connections because the buffer pool is bypassed and data is read directly from disk. It is recommended to only use `db2dart` with the database deactivated.

- **db2cklog**

  Use the `db2cklog` utility to query a transaction log. This utility is intrusive and should be issued on log files that are closed; such as archive log files.

- **db2diag**

  Use the `db2diag` utility to query diagnostic data. The `db2diag` utility is non-intrusive.

- **db2level**

  Use the db2level utility to display the DB2 software version level in use. This utility is non-intrusive.

- **db2look**

  Use the `db2look` utility to extract DDL (Data Definition Language) for objects in your database. The `db2look` utility is generally non-intrusive but can create locks and so should be issued during off-peak windows.

- **Inspect**

  Use the `INSPECT` utility to query table space data. The `INSPECT` utility is similar to the `db2dart` utility , but limited to a specified table space. The `INSPECT` utility is intrusive when used with the `rowcompestimate` parameter.

## *4.4 How are empty blocks in an MDC returned to the table space after a delete?*

**Answer**:

When a `DELETE` statement has removed rows in an MDC table, the empty blocks must be reclaimed since they are not automatically released to the table space. Use the `RECLAIM EXTENTS ONLY` option of the `REORG` command to reclaim the space without performing a full `REORG`.

## *4.5 How do I make database configuration changes in a high availability (HA) environment*

**Answer**:

If you need to restart the instance to effect a parameter change in HA controlled environment, use the `db2stop` command followed by the `db2start` command as per normal operations.

Issuing the `db2stop` command in a HA environment does not trigger a failover. However, it locks the `resourcegroup` until the `db2start` command is issued.

# Education: Training and references

Training and education roadmap for IBM Smart Analytics System and InfoSphere Balanced Warehouse: http://www.ibm.com/software/data/education/roadmaps.html#isas

## *Useful links*

- IBM Smart Analytics System external web site:

  http://www.ibm.com/software/data/infosphere/smart-analytics-system/

- Download site for IBM Smart Analytics System and InfoSphere Balanced Warehouse documentation:

  https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?lang=en_US&source=idwbcu

- IBM Smart Analytics System support portal:

  http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/IBM_Smart_Analytics_System

- InfoSphere Balanced Warehouse support portal:

  http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/InfoSphere_Balanced_Warehouse

- All IBM Smart Analytics System and InfoSphere Balanced Warehouse validated stack pages:

  http://www.ibm.com/support/docview.wss?uid=swg21429594

# Further Reading

- Information Management Best Practices

  http://www.ibm.com/developerworks/data/bestpractices/

- IBM Smart Analytics System Redbook

  http://www.redbooks.ibm.com/redbooks/pdfs/sg247908.pdf

- Frequently Asked Question on System Administration

  http://www.ibm.com/developerworks/data/bestpractices/db2_bestpractices_smartanalytics/index.html

# Contributors

**Damian Ali**

*Information Developer, Data Warehousing, IBM*

**Enzo Cialini**

*Senior Technical Staff Member, DB2 Distributed and Data Warehousing, IBM*

**Gregg Snodgrass**

*IBM Smart Analytics System Development*

**Joyce Coleman**

*IBM Smart Analytics Systems Information Development*

**Katherine Kurtz**

*Manager, DB2 LUW Information Development*

**Maksym Petrenko**

*IBM DB2 Warehouse Integration Specialist*

**Marco Norbiato**

*Advanced DB2 DBA, IBM*

**Nathan Gevaerd Colossi**

*IBM Smart Analytics System Solutions Development*

**Nela Krawez**

*IBM Smart Analytics System Solutions Development*

**Sam Sukjirawat**

*IBM Smart Analytics System Lab Services*

**Silvio Luiz Correia Ferrari**

*Senior IT Specialist*

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "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.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

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

Any references in this information 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 use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other

## Trademarks