

Db2 11.1 for Linux, UNIX, and Windows

Registry and Environment Variables



Db2 11.1 for Linux, UNIX, and Windows

Registry and Environment Variables

Notice regarding this document

This document in PDF form is provided as a courtesy to customers who have requested documentation in this format. It is provided As-Is without warranty or maintenance commitment.

Contents

Notice regarding this document iii
Figures
Tables
Registry and environment variables 1
Environment variables and the profile registries 1
Profile registry locations and authorization
requirements
Setting registry and environment variables 2
Setting environment variables outside the profile
registries on Windows
Setting environment variables outside the profile
registries on Linux and UNIX operating systems . 5
Identifying the current instance 6

Index	
Miscellaneous variables	
Performance variables	
Query compiler variables	
Db2 pureScale environment variables 45	
Partitioned database environment variables 43	
Command line variables 41	
Communications variables	
System environment variables	
General registry variables	
Db2 registry and environment variables 8	
Aggregate registry variables 7	
partitioned database environment 6	
Setting variables at the instance level in a	

Figures

Tables

1.	Profile registry locations and authorization	
	requirements	. 2
2.	Common commands for setting registry	
	variables.	. 3
3.	Registry and environment variables summary	9

4.	How Parallelism is Calculated	. 29
5.	Default <i>minfree</i> , <i>maxfree</i> settings	. 61
6.	Values to use for DB2_EVMON_STMT_FILTER to	
	restrict event monitor output to specific SQL	
	operations	. 84

Registry and environment variables

Environment variables and the profile registries

Environment and registry variables control your Db2[®] database environment. Use the Db2 profile registries to view and update information about variables and instances.

Before the Db2 database profile registries were introduced, setting environment variables required you to specify a value for an environment variable and restart your computer. You can now use the Db2 profile registries to control most variables that affect your Db2 database environment.

Use the profile registries to control the environment variables from one computer. Different levels of support are provided through the different profiles. You can administer the environment variables remotely by using the Db2 administration server.

A Db2 database is affected by the following profile registries:

- The Db2 instance-level profile registry contains registry variables for an instance. Values that are defined in this registry override their settings in the global registry.
- The Db2 global-level profile registry contains settings that are used if a registry variable is not set for an instance. All instances that pertain to a particular copy of Db2 Enterprise Server Edition can access this registry.
- The Db2 instance node-level profile registry contains variable settings that are specific to a database partition in a partitioned database environment. Values that are defined in this registry override their settings at the instance and global levels.
- The Db2 user-level profile registry contains settings that are specific to each user. Values that are defined in this registry override their settings in the other registries.

The Db2 database system configures the operating environment by checking for registry values and environment variables and resolving them in the following order:

- 1. Environment variables that are set outside the profile registries.
- 2. Registry variables that are set with the user-level profile.
- 3. Registry variables that are set with the instance node-level profile.
- 4. Registry variables that are set with the instance-level profile.
- 5. Registry variables that are set with the global-level profile.

The Db2 instance profile registry contains a list of all instances that are associated with the current copy. A list exists for each Db2 copy. You can see the complete list of all the instances that are available on the system by running the **db2ilist** command. This profile registry does not contain variable settings.

Profile registry locations and authorization requirements

The Db2 profile registries have different locations and authorization requirements on each operating system. Authorization is required to update the values of variables in each profile registry.

Profile registry	Location on Windows	Location on Linux and UNIX	Linux and UNIX authorization requirements	Windows authorization requirements
Instance-level profile registry	\HKEY_LOCAL_computer \SOFTWARE\IBM\DB2 \PROFILES\ instance_name	<pre>instance_home/sqllib/ profile.env where instance_home is the home path of the instance owner.</pre>	-rw-rw-r instance_owner instance_owner_group profile.env	You must be a member of the Db2 administrators group (DB2ADMNS).
Global-level profile registry	\HKEY_LOCAL_computer \SOFTWARE\IBM\DB2 \GLOBAL_PROFILE	For root installations:/var/db2/ global.reg For non-root installations: <i>home_directory</i> /sqllib /global.reg	To modify a global registry variable in root installations, you must be logged on with root authority.	You must be a member of the Db2 administrators group (DB2ADMNS).
Instance node-level profile registry	\SOFTWARE\IBM\DB2\ PROFILES \instance_name\NODES\ node_number	<i>instance_home</i> /sqllib/ nodes / <i>node_number</i> .env where <i>instance_home</i> is the home path of the instance owner.	For the directory that contains the file: drwxrwsr-w instance_owner instance_owner_group nodes For the file: -rw-rw-r instance_owner instance_owner group node_number.env	You must be a member of the Db2 administrators group (DB2ADMNS).
User-level profile registry	The Lightweight Directory Access Protocol (LDAP) directory.	Does not apply.	Does not apply.	You must be a member of the Db2 administrators group (DB2ADMNS).
Instance profile registry	\HKEY_LOCAL_computer \SOFTWARE\IBM\DB2\ PROFILES \instance_name	For root installations:/var/db2/ global.reg For non-root installations: <i>home_directory</i> /sqllib /global.reg	None required.	None required.

Table 1. Profile registry locations and authorization requirements

Setting registry and environment variables

Most environment variables are set in the Db2 database profile registries by using the **db2set** command. The few variables that are set outside the profile registries require different commands depending on your operating system.

Before you begin

Ensure that you have the privileges that are required to set registry variables.

On Linux and UNIX operating systems, you must have the following privileges:

• SYSADM authority to set variables in the instance-level registry

• root authority to set variables in the global-level registry

On Windows operating systems, you must have one of the following privileges:

- local Administrator authority
- SYSADM authority with the following conditions:
 - If extended security is enabled, SYSADM users must belong to the DB2ADMNS group.
 - If extended security is not enabled, SYSADM users can make updates if the appropriate permissions are granted to them in the Windows registry.

About this task

When you use the **db2set** command to set variables in the profile registries, you do not need to restart your computer for variable values to take effect. However, changes do not affect Db2 applications that are currently running or users that are active. The Db2 registry applies the updated information to Db2 server instances and Db2 applications that are started after the changes are made.

If Db2 variables are set outside the registry, you cannot administer those variables remotely. Also, you must restart the computer for the variable values to take effect.

The **DB2INSTANCE** and **DB2NODE** Db2 environment variables are not stored in the Db2 profile registries. See the topics about setting environment variables outside the profile registries for information about setting these variables.

On Linux and UNIX operating systems, the instance-level profile registry is stored in the profile.env text file. If two or more users set a registry variable with the **db2set** command at almost the same time, the size of this file is reduced to zero. Also, the output from the **db2set -all** command displays inconsistent values.

Procedure

To set a registry variable:

Issue the **db2set** command with the relevant parameters. The following table shows some of the ways that you can set registry variables with the **db2set** command. See the **db2set** command reference topic for more information about the parameters and usage of this command.

Table 2. Common commands for setting registry variables

Desired Action	Command
Set a registry variable for the current or default instance.	db2set registry_variable_name=new_value
Set a registry variable for all databases in an instance.	db2set registry_variable_name=new_value -i instance_name
Set a registry variable for a particular database partition in an instance.	db2set registry_variable_name=new_value -i instance_name database_partition_number
Set a registry variable for all instances that pertain to a Db2 Enterprise Server Edition installation.	db2set registry_variable_name=new_value -g
Set a registry variable at the user level in a Lightweight Directory Access Protocol (LDAP) environment.	db2set registry_variable_name=new_value -ul

Table 2. Common commands for	setting registry variables	(continued)
------------------------------	----------------------------	-------------

Desired Action	Command
Set a registry variable at the global level in an LDAP environment	db2set registry_variable_name=new_value _al
DB2LDAP_KEEP_CONNECTION and DB2LDAP_SEARCH_SCOPE are the only two registry variables that can be set at the	3.
LDAP global level.	

Tip: If a registry variable requires Boolean values as arguments, the values YES, 1, TRUE, and ON are all equivalent and the values NO, 0, FALSE, and OFF are also equivalent. For any variable, you can specify any of the appropriate equivalent values.

Setting environment variables outside the profile registries on Windows

On Windows operating systems, the **DB2INSTANCE**, **DB2NODE**, and **DB2PATH** environment variables can be set only outside the profile registries. You are required to set only the **DB2PATH** variable.

About this task

On Windows operating systems, the following environment variables are set outside the profile registries:

- The **DB2INSTANCE** environment variable specifies the instance that is active by default. If this variable is not set, the Db2 database manager uses the value of the **DB2INSTDEF** variable as the current instance.
- The **DB2NODE** environment variable specifies the target logical node of a database partition server to which requests are routed.
- The **DB2PATH** environment variable specifies the directory where the Db2 database product is installed on Windows 32-bit operating systems.

If you want to set any other variables, those variables must be set in one or more of the profile registries.

You can determine the value of an environment variable by using the **echo** command. For example, to check the value of the **DB2NODE** environment variable, issue the following command:

echo %db2path%

Procedure

To set an environment variable outside the profile registries:

Set an environment variable by using one of the following options.

Option	Description
Set the environment variable at the instance level.	 Follow the appropriate procedure for your Windows operating system. Restart your computer.

Option	Description
Set the environment variable for the current session.	Issue the following command: set env_variable_name=new_value
Set the environment variable for the	db2start Issue the following command:
current session for a C shell.	setenv env_variable_name new_value

Setting environment variables outside the profile registries on Linux and UNIX operating systems

On Linux and UNIX operating systems, you must set the **DB2INSTANCE** system environment variable outside the profile registries. If you want to set any other variables, those variables must be set in one or more of the profile registries.

About this task

You can use the scripts db2profile (for Bourne or Korn shell) and db2cshrc (for C shell) to set the **DB2INSTANCE** variable to an instance name that you specify. The scripts are in the *instance_home*/sqllib directory, where *instance_home* is the home directory of the instance owner.

Instance owners and users with SYSADM privileges can customize these scripts for all users of an instance. Alternatively, users can copy and customize a script, then invoke a script directly or add it to their .profile or .login files.

To set variables that are not supported by the Db2 database manager, define them in the userprofile and usercshrc script files. These files are also in the *instance_home/sqllib* directory.

Note: By default, environment variables that are not prefixed with 'db2' or 'DB2' will be filtered out by the **db2start** command. To allow other environment variables, you can use the *DB2ENVLIST* environment variable. For more information, see "System environment variables" on page 22.

Procedure

To set an environment variable outside the profile registries:

Set an environment variable by using one of the following methods:

Option	Description
Set the environment variable at the instance level for a Bourne or Korn shell.	Run the db2profile script.
Set the environment variable at the instance level for a C shell.	Run the db2cshrc script.
Set the environment variable for the current session for a Bourne shell.	Issue the following command: export env_variable_name=new_value
Set the environment variable for the current session for a C shell.	Issue the following command: setenv env_variable_name new_value
Set the environment variable for the current session for a Korn shell.	<pre>Issue the following command: environment_variable_name=new_value export environment_variable_name</pre>

Identifying the current instance

Most commands that you issue or configuration changes that you make apply by default to the current instance. You can identify the current instance by checking the values of certain environment variables.

About this task

When you run commands to start or stop the database manager for an instance, the database manager applies the command to the current instance. To determine the current instance, the database manager checks the values of certain environment variables in the following order:

- 1. The value of the DB2INSTANCE environment variable for the current session.
- 2. The value of the DB2INSTANCE system environment variable.
- 3. On Windows operating systems, the value of the DB2INSTDEF registry variable.

Procedure

To identify the current instance:

Check the value of the relevant environment variable.

Option	Description
View the value of the DB2INSTANCE environment variable for the current session.	Issue the following command: db2 get instance
View the value of the DB2INSTANCE system environment variable.	 On Windows operating systems, issue the following command: echo %DB2INSTANCE% On Linux and UNIX operating systems, issue the following command: echo \$DB2INSTANCE
View the value of the DB2INSTDEF registry variable.	Issue the following command: db2set DB2INSTDEF

Setting variables at the instance level in a partitioned database environment

In a partitioned database environment, the way that you set registry variables in the instance-level profile registry depends on your operating system.

About this task

On Linux and UNIX operating systems, the instance-level profile registry is stored in a text file in the sqllib directory. Because the sqllib directory is on a file system that is shared by all physical database partitions, you can set a registry variable from any database partition.

On Windows operating systems, the Db2 database manager stores the instance-level profile registry in the Windows registry. As a result, data is not shared across physical database partitions. To set a variable for all database partitions, you must use the **rah** command to ensure that the command that you

use to set the variable is run on all computers. If you set a registry variable from a database partition and do not use the **rah** command, the variable is set only for that database partition in the current instance.

You can also use the **DB2REMOTEPREG** registry variable to configure a computer that is not the instance owner to use the values of registry variables on the instance-owning computer.

Procedure

To set a registry variable for all database partitions of the current instance:

Issue the command for your operating system from any database partition.

- On Linux and UNIX operating systems, issue the following command: db2set registry_variable_name=new_value
- On Windows operating systems, issue the following command: rah db2set registry_variable_name=new_value

Aggregate registry variables

Use an aggregate registry variable to group several registry variables as a configuration that is identified by another registry variable name. Each registry variable that is part of the group has a predefined setting. The aggregate registry variable is given a value that is interpreted as declaring several registry variables.

The intention of an aggregate registry variable is to ease registry configuration for broad operational objectives.

The only valid aggregate registry variable is **DB2_WORKLOAD**.

Valid values for this variable are:

- 10
- ANALYTICS
- CM
- COGNOS_CS
- FILENET CM
- INFOR_ERP_LN
- MAXIMO
- MDM
- SAP
- TPM
- WAS
- WC
- WP

Any registry variable that is implicitly configured through an aggregate registry variable might also be explicitly defined. Explicitly setting a registry variable that was previously given a value through the use of an aggregate registry variable is useful when doing performance or diagnostic testing. Explicitly setting a variable that is configured implicitly by an aggregate is referred to as overriding the variable. If you attempt to modify an explicitly set registry variable by using an aggregate registry variable, a warning is issued and the explicitly set value is kept. This warning tells you that the explicit value is maintained. If the aggregate registry variable is used first and then you specify an explicit registry variable, a warning is not given.

When you query the aggregate registry variable, only the value assigned to that variable is shown. Most users should not care about the values for each individual variable.

The following example shows the interaction between using the aggregate registry variable and explicitly setting a registry variable. To control your database environment, you might set the **DB2_WORKLOAD** aggregate registry variable to SAP and override the **DB2_SKIPDELETED** registry variable to N0. By running the **db2set** command, you receive the following results:

```
DB2_WORKLOAD=SAP
DB2_SKIPDELETED=N0
```

In another situation, you might set **DB2ENVLIST**, set the **DB2_WORKLOAD** aggregate registry variable to SAP, and override the **DB2_SKIPDELETED** registry variable to N0. When you issue the **db2set** command, the registry variables that were configured automatically by setting the aggregate registry variable show the name of the aggregate displayed in square brackets, next to its value. The **DB2_SKIPDELETED** registry variable shows a N0 value, with [0] displayed next to its value.

When you no longer require the configuration that is associated with **DB2_WORKLOAD**, delete the implicit values of each registry variable in the group by deleting the value of the aggregate registry variable. Use the following command to delete the value of the **DB2_WORKLOAD** variable:

db2set DB2_WORKLOAD=

After deleting the **DB2_WORKLOAD** aggregate registry variable value, restart the database. After the database is restarted, the registry variables that were implicitly configured by the aggregate registry variable are no longer in effect.

Deleting the value of an aggregate registry variable does not delete the value for a registry variable that was set explicitly. It does not matter that the registry variable is a member of the group definition that was deleted. The explicit setting for the registry variable is maintained.

You might need to see the values for each registry variable that is a member of the **DB2_WORKLOAD** aggregate registry variable. For instance, you might want to see the values that would be used if you configured **DB2_WORKLOAD** to SAP. To find the values that would be used if **DB2_WORKLOAD=**SAP, run db2set -gd DB2_WORKLOAD=SAP.

Db2 registry and environment variables

Db2 database products provide a number of registry variables and environment variables that you might need to know about to get up and running.

To view a list of all supported registry variables, execute the following command: db2set -lr

You must set values for registry variables that you want to update before you execute the **db2start** command.

The following table lists all registry variables by category.

Variable category	Registry or environment variable name	
General	DB2ACCOUNT DB2BIDI DB2_CAPTURE_LOCKTIMEOUT DB2CODEPAGE DB2_COLLECT_TS_REC_INFO DB2_CONNRETRIES_INTERVAL DB2CONSOLECP DB2DBDFT DB2DISCOVERYTIME DB2_ENFORCE_MEMBER_SYNTAX DB2FODC DB2_FORCE_APP_ON_MAX_LOG	DB2GRAPHICUNICODESERVERDB2INCLUDE DB2INSTDEF DB2INSTOWNER DB2_LIC_STAT_SIZE DB2LOCALE DB2_MAX_CLIENT_CONNRETRIES DB2_MAX_GLOBAL_SNAPSHOT_SIZE DB2_OBJECT_TABLE_ENTRIES DB2_OBJECT_TABLE_ENTRIES DB2_SYSTEM_MONITOR_SETTINGS DB2_SYSTEM_MONITOR_SETTINGS DB2TERRITORY DB2_VIEW_REOPT_VALUES DB2_ONLINERECOVERY_WITH_UR_ACCESS
System environment	DB2_ALTERNATE_GROUP_LOOKUP DB2_CLI_DRIVER_INSTALL_PATH DB2CONNECT_ENABLE_EURO_CODEPAGE DB2CONNECT_IN_APP_PROCESS DB2_COPY_NAME DB2_CPU_BINDING DB2DBMSADDR DB2_DIAGPATH DB2DOMAINLIST DB2ENVLIST DB2INSTANCE DB2INSTPROF DB2LDAPSecurityConfig DB2LIBPATH DB2LOGINRESTRICTIONS	DB2NODE DB2OPTIONS DB2_PARALLEL_IO DB2PATH DB2_PMAP_COMPATIBILITY DB2PROCESSORS DB2RCMD_LEGACY_MODE DB2RESILIENCE DB2_RESTORE_GRANT_ADMIN_ AUTHORITIES DB2_SKIP_CHK_TMP DB2_SYMPHONY_WLM DB2SYSTEM DB2_SYMPHONY_WLM DB2SYSTEM DB2TMPDIR DB2_UPDDBCFG_SINGLE_DBPARTITION DB2_USE_PAGE_CONTAINER_TAG DB2WORKDIR DB2_WORKLOAD
Communica- tions	DB2_ALLOW_WLB_WITH_SEQUENCES DB2CHECKCLIENTINTERVAL DB2COMM DB2FCMCOMM DB2_FORCE_NLS_CACHE DB2_PMODEL_SETTINGS DB2RSHCMD DB2RSHTIMEOUT	DB2SORCVBUF DB2SOSNDBUF DB2TCP_CLIENT_CONTIMEOUT DB2TCP_CLIENT_KEEPALIVE_TIMEOUT DB2TCP_CLIENT_RCVTIMEOUT DB2TCPCONNMGRS DB2TCP_SERVER_KEEPALIVE_TIMEOUT
Command- line	DB2BQTIME DB2BQTRY DB2_CLP_EDITOR DB2_CLP_HISTSIZE	DB2_CLPPROMPT DB2IQTIME DB2RQTIME
Partitioned database environment	DB2CHGPWD_EEE DB2_FCM_SETTINGS DB2_FORCE_OFFLINE_ADD_PARTITION	DB2_NUM_FAILOVER_NODES DB2_PARTITIONEDLOAD_DEFAULT DB2PORTRANGE
Db2 pureScale [®] environment	DB2_DATABASE_CF_MEMORY DB2_SD_ALLOW_SLOW_NETWORK	DB2_MCR_RECOVERY_PARALLELISM_CAP

Table 3. Registry and environment variables summary

Variable category	Registry or environment variable name	
Query compiler	DB2_ANTIJOIN DB2_DEFERRED_PREPARE_SEMANTICS DB2_INLIST_TO_NLJN DB2_LIKE_VARCHAR DB2_MINIMIZE_LISTPREFETCH	DB2_NEW_CORR_SQ_FF DB2_OPT_MAX_TEMP_SIZE DB2_REDUCED_OPTIMIZATION DB2_SELECTIVITY DB2_SQLROUTINE_PREPOPTS
Performance	DB2_ALLOCATION_SIZE DB2_APM_PERFORMANCE DB2ASSUMEUPDATE DB2_AVOID_PREFETCH DB2_BACKUP_USE_DIO DB2BPVARS DB2CHKPTR DB2CHKSQLDA DB2_CREATE_INDEX_ALLOW_WRITE DB2_CREATE_INDEX_ALLOW_WRITE DB2_EXTENDE_OL_UNIQUE_INDEX_ACCESS DB2_EXTENDED_IO_FEATURES DB2_EXTENDED_OPTIMIZATION DB2_IO_PRIORITY_SETTING 57 DB2_KEEP_AS_AND_DMS_CONTAINERS_ OPEN DB2_LARGE_PAGE_MEM DB2_LOGGER_NON_BUFFERED_IO DB2MAXFSCRSEARCH DB2_MAX_INACT_STMTS DB2_MAX_NON_TABLE_LOCKS DB2_MAX_NON_TABLE_LOCKS DB2_MAX_NON_TABLE_LOCKS DB2_MAX_NON_TABLE_LOCKS DB2_MEM_TUNING_RANGE DB2_MAAP_READ	DB2_MMAP_WRITE DB2_NO_FORK_CHECK DB2NTMEMSIZE DB2NTNOCACHE DB2NTPRICLASS DB2NTWORKSET DB2_OVERRIDE_BPF DB2_PINNED_BP DB2PRIORITIES DB2_RCT_FEATURES DB2_RESOURCE_POLICY DB2_SELUDI_COMM_BUFFER DB2_SET_MAX_CONTAINER_SIZE DB2_SKIPDELETED DB2_SKIPINSERTED DB2_SKIPINSERTED DB2_SMS_TRUNC_TMPTABLE_THRESH DB2_SORT_AFTER_TQ DB2_SQLWORKSPACE_CACHE DB2_USE_ALTERNATE_PAGE_CLEANING DB2_USE_FAST_PREALLOCATION DB2_USE_FAST_LOG_PREALLOCATION DB2_USE_IOCP

Table 3. Registry and environment variables summary (continued)

Variable category	Registry or environment variable name	
Miscella-		NR2 HISTORY FILTERNR2 INNEY POTEREE DEFAILET
neous	DB2 ATS FNARI F	DB2_NISTORT_TIETERDB2_NDEX_TOTTREE_DELADET
neous	DB2AIITH	DB2LDAR_DASEBA
	DB2 BCKP TNCLUDE LOGS WARNING	DB2LDAP CLITENT PROVIDER
	DB2_BCKP_PAGE_VERIFICATION	
	DB2CLIINIPATH	DB2LDAP KEEP CONNECTION
	DB2 COMMIT ON EXIT	DB2LDAP SEARCH SCOPE
	DB2 COMMON APP DATA PATH	DB2 LOAD COPY NO OVERRIDE
	DB2 COMPATIBILITY VECTOR	DB2_LIMIT_FENCED_GROUP
	DB2 CREATE DB ON PATH	DB2LOADREC
	DB2 DDL SOFT INVAL	DB2LOCK TO RB
	DB2 DISABLE FLUSH LOG	DB2 MAX LOB BLOCK SIZE
	DB2_DISPATCHER_PEEKTIMEOUT	DB2_MEMORY_PROTECT
	DB2_DJ_INI	DB2_MIN_IDLE_RESOURCES
	DB2_DMU_DEFAULT	DB2_NCHAR_SUPPORT
	DB2_DOCHOST	DB2NOEXITLIST
	DB2_DOCPORT	DB2_NUM_CKPW_DAEMONS
	DB2DSDRIVER_CFG_PATH	DB2_OPTSTATS_LOG
	DB2DSDRIVER_CLIENT_HOSTNAME	DB2REMOTEPREG
	DB2_ENABLE_AUTOCONFIG_DEFAULT	DB2_RESOLVE_CALL_CONFLICT
	DB2_ENABLE_LDAP	DB2_RESTRICT_DDF
	DB2_EVMON_EVENT_LIST_SIZE	DB2_SAS_SETTINGS
	DB2_EVMON_STMT_FILTER	DB2SATELLITEID
	DB2_EXTSECURITY	DB2_SERVER_CONTIMEOUT
	DB2_FALLBACK	DB2_SERVER_ENCALG
	DB2_FMP_COMM_HEAPSZ	DB2SORT
	DB2_GRP_LOOKUP	DB2_STANDBY_ISO
	DB2_HADR_BUF_SIZE	DB2STMM
	DB2_HADR_NO_IP_CHECK	DB2_SYNC_RELEASE_LOCK_ATTRIBUTES
	DB2_HADR_PEER_WAIT_LIMIT	DB2_TRUNCATE_REUSESTORAGE
	DB2_HADR_REPLAY_ONLY_WINDOW_DIAGLEVEL	DB2_UTIL_MSGPATH
	DB2_HADR_ROS	DB2_XBSA_LIBRARY
	DB2_HADR_SORCVBUF	DB2_XSLT_ALLOWED_PATH
	DB2_HADR_SOSNDBUF	

Table 3. Registry and environment variables summary (continued)

General registry variables

You set general registry variables to control database behaviors, such as the time interval between consecutive connection retries. Some registry variables apply only to specific operating system environments.

DB2ACCOUNT

- Operating system: All
- Default: NULL
- This variable defines the accounting string that is sent to the remote host. Refer to the Db2 Connect User's Guide for details.

DB2BIDI

- Operating system: All
- Default: NO, Values: YES or NO
- This variable enables bidirectional support and the **DB2CODEPAGE** variable is used to declare the code page to be used.

DB2_CAPTURE_LOCKTIMEOUT

- Operating system: All
- Default: NULL, Values: ON or NULL
- This variable specifies to log descriptive information about lock timeouts at the time that they occur. The logged information identifies: the key applications involved in the lock contention that resulted in the lock timeout, the details about what these applications were running at the time of the lock timeout, and the details about the lock causing the contention. Information is captured for both the lock requestor (the application that received the lock timeout error) and the current lock owner. A text report is written and stored in a file for each lock timeout.

The files are created using the following naming convention: db2locktimeout.*par*.*AGENTID*.*yyyy-mm-dd-hh-mm-ss*, where *par* is the database partition number; *AGENTID* is the Agent ID; *yyyy-mm-dd-hh-mm-ss* is the timestamp consisting of the year, month, day, hour, minute and second. In non-partitioned database environments, *par* is set to 0.

The location of the file is based on the value set in the **diagpath** database configuration parameter. If **diagpath** is not set, then the file is located in one of the following directories:

- In Windows environments:

- If you do not set the **DB2INSTPROF** environment variable, information is written to *x*:*SQLLIB\DB2INSTANCE*, where *x* is the drive reference, *SQLLIB* is the directory that you specified for the **DB2PATH** registry variable, and *DB2INSTANCE* is the name of the instance owner.
- If you set the DB2INSTPROF environment variable, information is written to x:\DB2INSTPROF\DB2INSTANCE, where x is the drive reference, DB2INSTPROF is the name of the instance profile directory, and DB2INSTANCE is the name of the instance owner.
- If you set the **DB2INSTPROF** environment variable to a new location, you must ensure that it contains the appropriate files and folders to run the instance. This may require you to copy all of the files and folders from the previous location to the new location.
- In Linux and UNIX environments: information is written to INSTHOME/sqllib/db2dump, where INSTHOME is the home directory of the instance.

Delete lock timeout report files when you no longer need them. Because the report files are in the same location as other diagnostics logs, the Db2 system could shutdown if the directory is allowed to get full. If you need to keep some lock timeout report files, move them to a directory or folder different than where the Db2 logs are stored.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

Important: This variable is deprecated and might be removed in a future release because there are new methods to collect lock timeout events using the CREATE EVENT MONITOR FOR LOCKING statement.

DB2CODEPAGE

- Operating system: All
- Default: derived from the language ID, as specified by the operating system.

• This variable specifies the code page of the data presented to Db2 for database client application. The user should not set **DB2CODEPAGE** unless explicitly stated in Db2 documents, or asked to do so by Db2 service. Setting **DB2CODEPAGE** to a value not supported by the operating system can produce unexpected results. Normally, you do not need to set **DB2CODEPAGE** because Db2 automatically derives the code page information from the operating system.

Note: Because Windows does not report a Unicode code page (in the Windows regional settings) instead of the ANSII code page, a Windows application will not behave as a Unicode client. To override this behavior, set the **DB2C0DEPAGE** registry variable to 1208 (for the Unicode code page) to cause the application to behave as a Unicode application.

DB2_COLLECT_TS_REC_INFO

- Operating system: All
- Default: ON; OFF for HADR databases, Values: ON or OFF
- This variable specifies whether all log files are to be processed when rolling forward a table space, regardless of whether the log files contain log records that affect the table space. To skip the log files known not to contain any log records affecting the table space, set this variable to ON.
 DB2_COLLECT_TS_REC_INFO must be set before the log files are created and used so that the information required for skipping log files is collected.

DB2_CONNRETRIES_INTERVAL

- Operating system: All
- Default: Not set, Values: an integer number of seconds
- This variable specifies the sleep time between consecutive connection retries, in seconds, for the automatic client reroute feature. You can use this variable in conjunction with DB2_MAX_CLIENT CONNRETRIES to configure the retry behavior for automatic client reroute.

If DB2_MAX_CLIENT_CONNRETRIES is set, but DB2_CONNRETRIES_INTERVAL is not, DB2_CONNRETRIES_INTERVAL defaults to 30. If

DB2_MAX_CLIENT_CONNRETRIES is not set, but DB2_CONNRETRIES_INTERVAL is set, DB2_MAX_CLIENT_CONNRETRIES defaults to 10. If neither DB2_MAX_CLIENT_CONNRETRIES nor DB2_CONNRETRIES_INTERVAL is set, the

automatic client reroute feature reverts to its default behavior of retrying the connection to a database repeatedly for up to 10 minutes.

DB2CONSOLECP

- Operating system: Windows
- Default: NULL, Values: all valid code page values
- Specifies the code page for displaying Db2 message text. When specified, this value overrides the operating system code page setting.

DB2DBDFT

- Operating system: All
- Default: NULL
- This variable specifies the database alias name of the database to be used for implicit connects. If an application has no database connection but SQL or XQuery statements are issued, an implicit connect will be made if the **DB2DBDFT** environment variable has been defined with a default database.

DB2DISCOVERYTIME

- Operating system: Windows
- Default: 40 seconds, Minimum: 20 seconds
- This variable specifies the amount of time that SEARCH discovery will search for Db2 systems.

DB2_ENFORCE_MEMBER_SYNTAX

- Operating system: All
- Default: OFF , Values: OFF or ON
- This variable allows you to control whether or not the syntax for SQL statements, Db2 commands, and APIs will be checked for the correct usage of the database partition keywords to determine whether the MEMBER keyword must be used instead. In a Db2 pureScale environment, the default behavior is to tolerate the usage of keywords specific to database partitions, such as DBPARTITIONNUM or DATABASE PARTITION, even when the operation is targeting a Db2 member.
- However, when **DB2_ENFORCE_MEMBER_SYNTAX** is set to ON, the MEMBER keyword must be specified correctly, otherwise SQL1538N is returned. The setting of this variable is ignored and has no effect outside of a Db2 pureScale environment.

DB2_EXPRESSION_RULES

- Operating system: All
- Default: Empty, Values: RAISE_ERROR_PERMIT_SKIP or RAISE_ERROR_PERMIT_DROP
- The settings for the **DB2_EXPRESSION_RULES** registry variable control how the Db2 Optimizer determines the access plan for queries which involve a RAISE_ERROR function. The default behaviour of the RAISE_ERROR function is that no filtering may be pushed beyond the expression containing this function. This can result in no predicates being applied during the table accesses which can lead to excessive computation of expressions, excessive locking and poor query performance.

In certain cases this behaviour is too strict, depending on the particular business requirements of the application, it may not matter if predicates and joins are applied before the application of RAISE_ERROR. For example in the context of a row level security implementation, there is typically an expression of the form:

CASE WHEN <conditions for validatin access to this row>

THEN NULL ELSE RAISE_ERROR(...)

END

The application may only be concerned with validating access to the rows which are selected by the query and not in validating access to every row in the table. Thus predicates could be applied in the base table access and the expression containing the RAISE_ERROR only needs to executed after all the filtering is performed. In this case a value of **DB2_EXPRESSION_RULES**=RAISE_ERROR_PERMIT_SKIP may be appropriate.

Another alternative is in the context of COLUMN LEVEL security. In this case there are typically expressions of the form:

CASE WHEN <conditions for validating access to this row and column> THEN <table.column>

ELSE RAISE_ERROR(...)

In this case the application may only want errors to be raised if the user attempts to receive the data for a particular row and column contains a value that the user is not allowed to retrieve. In this case a setting of **DB2_EXPRESSION_RULES**=RAISE_ERROR_PERMIT_DROP will only cause the expression containing the RAISE_ERROR function to be evaluated if the particular column is used by a predicate or a column function, or if it is returned as output from the query.

DB2F0DC

- Operating system: All
- Default: The concatenation of all FODC parameters (see following list)
 - for Linux and UNIX: "CORELIMIT=val DUMPCORE=ON DUMPDIR=diagpath"
 - for Windows: "DUMPDIR=diagpath"

Note that the parameters are separated by spaces.

• This registry variable controls a set of troubleshooting-related parameters used in First Occurrence Data Collection (FODC). Use **DB2FODC** to control different aspects of data collection in outage situations. The **DB2FODC** registry variable should be set at the instance level only.

This registry variable is read once, during the Db2 instance startup. To perform updates to the FODC parameters online, use **db2pdcfg** tool. Use the **DB2FODC** registry variable to sustain the configuration across reboots. You do not need to specify all of the parameters, nor do you need to specify them in a particular order. The default value is assigned to any parameter that is not specified. For example, if you don't want the core files dumped, but you do want the other parameters' default behaviors, you would issue the command:

db2set DB2FODC="DUMPCORE=OFF"

Parameters:

CORELIMIT

- Operating system: Linux and UNIX
- Default: Current[®] ulimit setting, Values: 0 to unlimited
- This option specifies the maximum size, in bytes, of core files created. This value overrides the current core file size limit setting. Consideration should be given to the available file system space because core files can be quite large. The size is dependent on the Db2 configuration and the state of the process at the time the problem occurs.

If **CORELIMIT** is set, Db2 will use this value to override the current user core limit (ulimit) setting to generate the core file.

If **CORELIMIT** is not set, Db2 will set the core file size to the value equal to the current ulimit setting.

Note: Any changes to the user core limit or **CORELIMIT** are not effective until the next recycling of the Db2 instance.

COS

- Operating system: All
- Default: ON, Values: ON or OFF
- This option specifies if the db2cos script is enabled or not.
 You can use the following parameters with this parameter:

COS_SLEEP

- Default: 3, Values: 0 to unlimited
- This option specifies the amount of time to sleep in seconds between checking the size of the output file generated.

COS_TIMEOUT

- Default: 30, Values: 0 to unlimited
- This option specifies the amount of time to wait in seconds before the script is finished.

COS_COUNT

- Default: 255, Values: 0 to 255
- This option specifies the number of times to execute **db2cos** during a database manager trap.

COS_SQLO_SIG_DUMP

- Default: ON, Values: ON or OFF
- This option specifies if **db2cos** is enabled when the **SQL0_SIG_DUMP** signal is received.

DUMPCORE

- Operating system: Linux, Solaris, AIX[®]
- Default: AUTO, Values: AUTO, ON, or OFF
- This option specifies if core file generation is to take place. Core files, which are used for problem determination and are created in the **diagpath** directory, contain the entire process image of the terminating Db2 process. However, whether or not an actual core file dump occurs depends on the current ulimit setting and value of the **CORELIMIT** parameter. Some operating systems also have configuration settings for core dumps, which may dictate the behavior of application core dumping. The AUTO setting causes a core file to be generated if a trap cannot be sustained when the **DB2RESILIENCE** registry variable is set to ON. The **DUMPCORE**=ON setting always generates a core file by overriding the **DB2RESILIENCE** registry variable setting.

The recommended method for disabling core file dumps is to set **DUMPCORE** to OFF.

DUMPDIR

- Operating system: All
- Default: **diagpath** directory, or the default diagnostic directory if **diagpath** is not defined, Values: *path to directory*
- This option specifies the absolute path name of the directory for core file creation.

FODCPATH

- Operating system: All
- Default: path defined by the **DIAGPATH** database manager configuration parameter, Values: *fodc_path_name*
- This option specifies the absolute path name of where the FODC package is to be directed. The *fodc_path_name* must be

an existing directory and must be writable by the member or members for which it is set for and by the fmp processes running on those members.

SERVICELEVEL

- Operating system: All
- Default: AUTOMATIC ulimit setting, Values: AUTOMATIC, BASIC, or FULL
- This option specifies how data is collected during panics, traps, or errors that might indicate data corruption. Db2 is designed to generate diagnostics that are appropriate to the configuration and problem context. For example, when a trap can be sustained, only the minimum essential diagnostics are generated in order to rollback the transaction and respond to the application as soon as possible, releasing resources which other applications may be waiting on. When a trap cannot be sustained, diagnostics such as db2cos data collection scripts and core dumps may be limited in favor of availability in Db2 pureScale configurations. The default behaviour for generating diagnostics is represented by the SERVICELEVEL setting of AUTOMATIC.

The following option are supported for this parameter:

AUTOMATIC

This setting specifies that the effective SERVICELEVEL setting (that is, BASIC or FULL) is to be chosen at runtime, for the members, and at start time, for the CF process. At present, the only times that BASIC is chosen are for Db2 pureScale environments that have multiple members and for trap resilience.

- **BASIC** This SERVICELEVEL setting specifies that a minimal amount of FODC data is to be dumped. Core dump processing is disabled by default (but can be overridden by the COREDUMP setting), diagnostics are restricted to the affected thread only, and callout scripts are disabled.
- **FULL** This SERVICELEVEL setting specifies that the maximum amount of FODC data is to be dumped. This includes core dumps, any associated components dumps, and the invocation of the callout scripts. In addition, there is no attempt to sustain traps.

DB2_FORCE_APP_ON_MAX_LOG

- Operating system: All
- Default: TRUE, Values: TRUE or FALSE
- Specifies what happens when the **max_log** configuration parameter value is exceeded. If set to TRUE, the application is forced off the database and the unit of work is rolled back.

If FALSE, the current statement fails. The application can still commit the work completed by previous statements in the unit of work, or it can roll back the work completed to undo the unit of work.

Note: This Db2 registry variable affects the ability of the import utility to recover from log full situations. If **DB2_FORCE_APP_ON_MAX_LOG** is set to

TRUE and you issue an **IMPORT** command with the **COMMITCOUNT** command option, the import utility will not be able to perform a commit in order to avoid running out of active log space. When the import utility encounters an SQL0964C (Transaction Log Full), it will be forced off the database and the current unit of work will be rolled back.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2GRAPHICUNICODESERVER

- Operating system: All
- Default: OFF, Values: ON or OFF
- This registry variable is used to accommodate existing applications written to insert graphic data into a Unicode database. Its use is only needed for applications that specifically send sqldbchar (graphic) data in Unicode instead of the code page of the client. (sqldbchar is a supported SQL data type in C and C++ that can hold a single double-byte character.) When set to ON, you are telling the database that graphic data is coming in Unicode, and the application expects to receive graphic data in Unicode.

DB2INCLUDE

- Operating system: All
- Default: Not set
- Specifies a path to be used during the processing of the SQL INCLUDE text-file statement during **DB PREP** processing. It provides a list of directories where the INCLUDE file might be found. Refer to Developing Embedded SQL Applications for descriptions of how **DB2INCLUDE** is used in the different precompiled languages.
- If not explicitly set, SQL INCLUDE ext-file statement will search text-file in the current directory.

DB2INSTDEF

- Operating system: All
- Default: Db2 on Windows, and the last instance that is created on UNIX.
- This variable sets the value to be used if **DB2INSTANCE** is not defined.

DB2INSTOWNER

- Operating system: Windows
- Default: NULL
- The registry variable created in the Db2 profile registry when the instance is first created. This variable is set to the name of the instance-owning machine.

DB2_LIC_STAT_SIZE

- Operating system: All
- Default: NULL, Range: 0 to 32767
- This variable determines the maximum size (in MBs) of the file containing the license statistics for the system. A value of zero turns the license statistic gathering off. If the variable is not recognized or not defined, the variable defaults to unlimited. The statistics are displayed using the License Center.

DB2LOCALE

- Operating system: All
- Default: NO, Values: YES or NO
- This variable specifies whether the default "C" locale of a process is restored to the default "C" locale after calling Db2 and whether to restore the process locale back to the original 'C' after calling a Db2 function. If the original locale was not 'C', then this registry variable is ignored.

DB2_MAX_CLIENT_CONNRETRIES

- Operating system: All
- Default: Not set, Values: an integer number of maximum times to retry the connection
- This variable specifies the maximum number of connection retries that the automatic client reroute feature will attempt. You can use this variable in conjunction with DB2_CONNRETRIES_INTERVAL to configure the retry behavior for automatic client reroute.

If DB2_MAX_CLIENT_CONNRETRIES is set, but DB2_CONNRETRIES_INTERVAL is not, DB2_CONNRETRIES_INTERVAL defaults to 30. If

DB2_MAX_CLIENT_CONNRETRIES is not set, but DB2_CONNRETRIES_INTERVAL is set, DB2_MAX_CLIENT_CONNRETRIES defaults to 10. If neither

DB2_MAX_CLIENT_CONNRETRIES nor **DB2_CONNRETRIES_INTERVAL** is set, the automatic client reroute feature reverts to its default behavior of retrying the connection to a database repeatedly for up to 10 minutes.

DB2_MAX_GLOBAL_SNAPSHOT_SIZE

- Operating system: All
- Default: Not set, Values: 0 to the maximum size of a snapshot.
- This variable specifies the number of bytes a snapshot or snapshot estimate can be. You can use this variable to prevent large global snapshots from causing memory usage spikes which can cause performance degradation and system hangs.

By default, **DB2_MAX_GLOBAL_SNAPSHOT_SIZE** is not set, which means an effective limit of the maximum size of a snapshot (2 GB less 512 bytes). This variable is dynamic and only applies to partitioned database environments.

DB2_OBJECT_TABLE_ENTRIES

- Operating system: All
- Default: 0, Values: 0-65532

The actual maximum value possible on your system depends on the page size and extent size, but it cannot exceed 65532.

• This variable specifies the expected number of objects in a table space. If you know that a large number of objects (for example, 1000 or more) will be created in a DMS table space, you should set this registry variable to the approximate number before creating the table space. This will reserve contiguous storage for object metadata during table space creation. Reserving contiguous storage reduces the chance that an online backup will block operations which update entries in the metadata (for example, CREATE INDEX, **IMPORT REPLACE**). It will also make resizing the table space easier because the metadata will be stored at the start of the table space.

If the initial size of the table space is not large enough to reserve the contiguous storage, the table space creation will continue without the additional space reserved.

DB2_SRVLSTLOG_LEVEL

- Operating system: All
- Default: 1, Values: 0-4
- Specifies the logging level for server list events that pertain to workload balancing (WLB) and automatic client reroute (ACR). You can use this information (usually under the guidance of IBM[®] service) to gather problem determination data. All entries that are logged are informational. Valid values for this registry variable are as follows:
 - 0: Nothing is logged
 - 1: Only messages of high importance are logged.
 - 2: Only messages of medium and high importance are logged.
 - 3: Only messages of low, medium, and high importance are logged.
 - 4: All messages are logged.

The **diagpath** database manager configuration parameter specifies where the server list log files are to be stored. These log files are circular and use the following naming convention: db2srvlst.0.log, db2srvlst.1.log, db2srvlst.N.log. Changes to **DB2_SRVLSTLOG_LEVEL** require the client application to be restarted before the new value takes effect.

DB2_SYSTEM_MONITOR_SETTINGS

- Operating system: All
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.
- The registry variable controls a set of parameters which allow you to modify the behavior of various aspects of Db2 monitoring. Separate each parameter by a semicolon, as in the following example:

db2set DB2_SYSTEM_MONITOR_SETTINGS=OLD_CPU_USAGE:TRUE; DISABLE_CPU_USAGE:TRUE

Every time you set **DB2_SYSTEM_MONITOR_SETTINGS**, each parameter must be set explicitly. Any parameter that you do not specify when setting this variable reverts back to its default value. So in the following example:

db2set DB2_SYSTEM_MONITOR_SETTINGS=DISABLE_CPU_USAGE:TRUE

OLD_CPU_USAGE will be restored to its default setting.

Note: Currently, this registry variable only has settings for Linux; additional settings for other operating systems will be added in future releases.

• Parameters:

OLD_CPU_USAGE

- Operating system: Linux
- Values: TRUE/ON, FALSE/OFF
- Default value on RHEL4 and SLES9: TRUE (Note: a setting of FALSE for OLD_CPU_USAGE will be ignored-only the old behavior will be used.)
- Default value on RHEL5, SLES10, and others: FALSE

- This parameter controls how the instance obtains CPU usage times on Linux platforms. If set to TRUE, the older method of getting CPU usage time is used. This method returns both system and user CPU usage times, but consumes more CPU in doing so (that is, it has a higher overhead). If set to FALSE, the newer method of getting CPU usage is used. This method returns only the user CPU usage value, but is faster because it has less overhead.

DISABLE_CPU_USAGE

- Operating system: Linux
- Values: TRUE/ON, FALSE/OFF
- Default value on RHEL4 and SLES9: TRUE
- Default value on RHEL5, SLES10, and others: FALSE
- This parameter allows you to determine whether CPU usage is read or not. When DISABLE_CPU_USAGE is enabled (set to TRUE), CPU usage is not read, allowing you to avoid the overhead that can sometimes occur during the retrieval of CPU usage.

DB2TERRITORY

- Operating system: All
- Default: derived from the language ID, as specified by the operating system.
- This variable specifies the region, or territory code of the client application, which influences date and time formats.

DB2_VIEW_REOPT_VALUES

- Operating system: All
- Default: NO, Values: YES, NO
- This variable enables all users to store the cached values of a reoptimized SQL or XQuery statement in the EXPLAIN_PREDICATE table when the statement is explained. When this variable is set to N0, only DBADM is allowed to save these values in the EXPLAIN_PREDICATE table.

DB2_ONLINERECOVERY

- Operating system: All
- Default: NO, Values: YES, NO
- This variable specifies whether database connectivity is allowed during the backward (transaction undo) phase of crash recovery. For more information, see "Database accessibility during backward phase of crash recovery or HADR takeover".
- This variable is available in Version 11.1.2.2 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

DB2_ONLINERECOVERY_WITH_UR_ACCESS

- Operating system: All
- Default: YES (NO when DB2_WORKLOAD=SAP), Values: YES, NO
- If the database is configured to allow for connectivity during the backward phase of crash recovery (when DB2_ONLINERECOVERY is set to YES), then this variable controls table accessibility during crash recovery. If this variable is set to YES, then the tables that are involved in crash

recovery might be accessible to SQL queries with UR isolation level. If this variable is set to N0, then the tables involved in crash recovery will not be accessible to SQL queries. Tables that are not involved in crash recovery will always be accessible to SQL queries in any isolation level. For more information, see "Database accessibility during backward phase of crash recovery or HADR takeover".

- This variable is available in Version 11.1.2.2 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

System environment variables

You use the system environment variables to pass configuration values to running applications in the Db2 environment. Some system environment variables only apply to specific operating system environments.

DB2_ALTERNATE_GROUP_LOOKUP

- Operating system: AIX, Linux
- Default: NULL, Values: NULL, GETGRSET on AIX, GETGROUPLIST on Linux
- This variable allows Db2 database systems to obtain group information from an alternative source provided by the operating system. On AIX, the function getgrset is used. This provides the ability to obtain groups from somewhere other than local files via Loadable Authentication Modules.

DB2_APPL_CFG_PATH

Operating system: Linux and UNIX

Note: The **DB2_APPL_CFG_PATH** environment variable is not supported on Linux and UNIX servers.

- Default: \$DB2_NET_CLIENT_PATH
- This variable specifies the path where the client configuration files are stored. The client configuration files include the client database manager configuration file, Db2 registry variables, directory catalogs, and the db2cli.ini file.
- You can set this variable to specify a user-specific client configuration path. If you set the DB2_APPL_CFG_PATH variable to a user-specific path, any changes that are made to the files in default location do not affect that client.
- There are other environment variables to override the default location of certain client configuration files, such as the db2cli.ini and db2dsdriver.cfg files. If these other environment variables are set, the DB2_APPL_CFG_PATH variable is ignored.

DB2_APPL_DATA_PATH

Operating system: Linux and UNIX

Note: The **DB2_APPL_DATA_PATH** environment variable is not supported on Linux and UNIX servers.

- Default: \$DB2_NET_CLIENT_PATH
- This variable specifies the path where the client user data is stored. Client user data includes the db2dump directory, CLI internal cache files, diagnostic data files, server list cache files, and internal temporary files that require write access.

- Client users cannot share application data paths.
- If the **DB2_NET_CLIENT_PATH** variable is set to a read only network path, you must set this variable to a location that you have write access to.
- If you set the **DIAGPATH** configuration parameter and the **DIAGPATH** keyword in the db2cli.ini file, the **DB2_APPL_DATA_PATH** variable is ignored.
- When **DB2_APPL_DATA_PATH** is set to an invalid or non-writeable path, a SQL1042C error might occur.

DB2_CLI_DRIVER_INSTALL_PATH

- Operating system: All
- Default: NULL
- This environment variable specifies the IBM Data Server Driver for ODBC and CLI installation directory. If there are multiple IBM Data Server Driver for ODBC and CLI installations in an environment, the value of the DB2_CLI_DRIVER_INSTALL_PATH environment variable therefore determines which copy of the IBM Data Server Driver for ODBC and CLI installation applications can use.

DB2_CLP_EDITOR

See DB2_CLP_EDITOR in "Command-line variables" for details.

DB2_CLP_HISTSIZE

See DB2_CLP_HISTSIZE in "Command-line variables" for details.

DB2CONNECT_ENABLE_EURO_CODEPAGE

- Operating system: All
- Default:NO, Values: YES or NO
- Set this variable to YES on all Db2 Connect clients and servers that connect to a Db2 for z/OS[®] server or a Db2 for IBM i server where euro support is required. If you set this variable to YES, the current application code page is mapped to the equivalent coded character set ID (CCSID) that explicitly indicates support for the euro sign.
- As a result, Db2 Connect connects to the Db2 for z/OS server or Db2 for IBM i server by using a CCSID that is a superset of the CCSID of the current application code and that also supports the euro sign. For example, if the client is using code page that maps to CCSID 1252, the client connects by using CCSID 5348.

DB2CONNECT_IN_APP_PROCESS

- Operating system: All
- Default: YES, Values: YES or NO
- When you set this variable to N0, local Db2 Connect clients on a Db2 Enterprise Server Edition machine are forced to run within an agent. Some advantages of running within an agent are that local clients can be monitored and that they can use SYSPLEX support.

DB2_COPY_NAME

- Operating system: Windows
- Default: The name of the default copy of Db2 installed on your machine. Values: the name of a copy of Db2 installed on your machine. The name can be up to 128 characters long.
- The DB2_COPY_NAME variable stores the name of the copy of Db2 currently in use. If you have multiple Db2 copies installed on your machine, you cannot use DB2_COPY_NAME to switch to a different copy of Db2, you must

run the command *INSTALLPATH*\bin\db2envar.bat to change the copy currently in use, where *INSTALLPATH* is the location where the Db2 copy is installed.

DB2_CPU_BINDING

- Operating system: Linux
- Default: AUTOMATIC
 - If a Db2 member and cluster caching facility (CF) are on the same host:
 - For the member, NUM_CORES=max(1, floor(0.8*totalCores))
 - For the cluster caching facility, NUM_CORES=totalCores the number listed previously.
 - If a Db2 member and cluster caching facility are not sharing a host, this variable is not set
- This registry variable controls CPU pinning. For changes to this variable to take effect, you need to restart the Db2 instance.

Parameters:

NUM_CORES

- Operating system: Linux
- Default: If the member or CF are on the same host, approximately 80% of the total available cores are assigned to Db2 and the remainder is assigned to the CF. Values: 0 < x < (number of physical cores on the host)
- This option specifies the number of cores to which the member or CF processes are pinned. You can use NUM_CORES to configure sub-capacity licensing of the Db2 product. The number of cores can be a whole number or a fraction, which allows you to add one or more hardware threads if simultaneous multithreading (SMT) is enabled.

PROCESSOR_LIST

- Operating system: Linux
- Default: Not set, Values: any of the processor numbers
- This option specifies which logical processors Db2 will be bound to, giving you complete control over the number of logical processors (or cores) and which CPU package (or socket) they will reside on. If you try to set both PROCESSOR_LIST and NUM_CORES with DB2_CPU_BINDING, NUM_CORES is ignored

MACHINE_SHARE

- Operating system: All
- Default: Not set, Values: $0 < x \le 100$, where x is an integer
- This option specifies the percentage of processors on the machine that Db2 should be bound to. If you have heterogenous machine hardware configurations in a Db2 instance, use it to set the percentage consumption of the total CPU processing on the machine.

Restart light considerations

If a member is restarted as a guest member on a host which already has a member running on it, the restart light member
will be pinned to the cores already being used by the resident member, up to the number of cores specified by **DB2_CPU_BINDING**.

If a member is restarted as a guest member on a host with fewer cores than what is specified by **DB2_CPU_BINDING**, the member will be bound to the number of cores on the host.

Every time you set **DB2_CPU_BINDING**, any parameter that is not explicitly set is cleared in the instance-level profile. Enclose each parameter and its value in quotes, as shown in the following examples.

Example 1

A user wants to pin the first member (which has an ID of 0) of Db2 instance db2inst1 to one core on a host machine with two cores:

db2set -i db2inst1 0 DB2_CPU_BINDING="NUM_CORES=1"

Example 2

A user wants to bind all the members in db2inst1 to five logical processors on a host machine with eight cores and Intel HTT enabled (meaning it has 16 logical processors):

db2set -i db2inst1 DB2_CPU_BINDING="NUM_CORES=2.5"

Example 3

A user wants to specify how many cores the primary CF (which has an ID of 128) is bound to:

db2set -i db2inst1 128 DB2 CPU BINDING="NUM CORES=4"

Example 4

A user wants to bind Db2 for db2inst1 on member 0 to a specific group of logical processors:

db2set -i db2inst1 0 DB2_CPU_BINDING="PROCESSOR_LIST=2,10,6,14"

DB2DBMSADDR

- Operating system: Linux on x86, Linux on zSeries (31-bit), and Windows 32-bit
- Default: NULL on Linux operating systems, 0x20000000 on Windows operating systems, Values: virtual addresses in the range 0x09000000 to 0xB0000000 in increments of 0x10000 on Linux operating systems, 0x20000000 to 0xB0000000 in increments of 0x10000 on Windows operating systems
- The **DB2DBMSADDR** registry variable specifies the default database shared memory address in hexadecimal format.

This variable can be used to fine-tune the address space layout of Db2 processes. This variable changes the location of the instance shared memory from its current location at virtual address 0x10000000 to the new value.

Note:

An incorrect address can cause severe issues with the Db2 database system, ranging from an inability to start a Db2 instance, to an inability to connect to the database. An incorrect address is one that collides with an area in memory that is already in use, or is predestined to be used for something else. To address this problem, reset the **DB2DBMSADDR** registry variable to NULL by using the following command: db2set DB2DBMSADDR= **Note:** Before changing the setting of this variable, you must stop the instance and all Db2 processes. If the instance is running while this variable is set, then any subsequent **db2stop** command will fail.

DB2_DIAGPATH

- Operating system: All
- Default: The default value is the instance db2dump directory on UNIX and Linux operating systems, and the instance db2 directory on Windows operating systems.
- This parameter applies to ODBC and CLI applications only.

This parameter allows you to specify the fully qualified path for Db2 diagnostic information. This directory could possibly contain dump files, trap files, an error log, a notification file, and an alert log file, depending on your platform.

Setting this environment variable has the same effect for ODBC and CLI applications in the scope of that environment as setting the Db2 database manager configuration parameter **diagpath**, and as setting the CLI/ODBC configuration keyword **DiagPath**.

DB2D0MAINLIST

- Operating system: All
- Default: NULL, Values: A list of Windows domain names separated by commas (",").
- This variable defines one or more Windows domains. The list, which is maintained on the server, defines the domains that the requesting user ID is authenticated against. Only users belonging to these domains have their connection or attachment requests accepted.

This variable is effective only when CLIENT authentication is set in the database manager configuration. It is needed if a single sign-on from a Windows desktop is required in a Windows domain environment.

DB2DOMAINLIST is supported if either the client or the server is running in a Windows environment.

DB2ENVLIST

- Operating system: UNIX
- Default: NULL
- This variable lists specific variable names for either stored procedures or user-defined functions. By default, the db2start command filters out all user environment variables except those prefixed with "Db2" or "db2". If specific environment variables must be passed to either stored procedures or user-defined functions, you can list the variable names in the DB2ENVLIST environment variable. Separate each variable name by one or more spaces.

DB2INSTANCE

- Operating system: All
- Default: DB2INSTDEF on Windows 32-bit operating systems.
- This environment variable specifies the instance that is active by default. On UNIX, users must specify a value for **DB2INSTANCE**.

Note: You cannot use the **db2set** command to update this registry variable. For more information, see "Identifying the current instance" on page 6 and "Setting environment variables outside the profile registries on Windows" on page 4.

DB2INSTPROF

- Operating system: Windows
- Default: ProgramData\IBM\DB2\Copy Name
- This environment variable specifies the location of the instance directory on Windows operating systems. The instance directory (and other user data files) cannot be under the sqllib directory.

DB2LDAPSecurityConfig

- Operating system: All
- Default: NULL, Values: valid name and path to the IBM LDAP security plug-in configuration file
- This variable is used to specify the location of the IBM LDAP security plug-in configuration file. If the variable is not set, the IBM LDAP security plug-in configuration file is named IBMLDAPSecurity.ini and is in one of the following locations:
 - On Linux and UNIX operating systems: INSTHOME/sqllib/cfg/
 - On Windows operating systems: %DB2PATH%\cfg\

On Windows operating systems, this variable should be set in the global system environment to ensure it is picked up by the Db2 service.

DB2LIBPATH

- Operating system: UNIX
- Default: NULL
- Db2 constructs its own shared library path. If you want to add a PATH into the engine's library path (for example, on AIX, a user-defined function requires a specific entry in LIBPATH), you must set DB2LIBPATH. The actual value of DB2LIBPATH is appended to the end of the Db2 constructed shared library path.

DB2LOGINRESTRICTIONS

- Operating system: AIX
- Default: LOCAL, Values: LOCAL, REMOTE, SU, NONE
- This registry variable allows you to use an AIX operating system API called loginrestrictions(). This API determines whether a user is allowed to access the system. By calling this API, Db2 database security is able to enforce the login restrictions that are specified by the operating system. There are different values that can be submitted to this API when you are using this registry variable. The values are:
 - REMOTE

Connection to Db2 is denied to users who are denied AIX login by rlogin or telnet. This option is equivalent to the S_RLOGIN option of the loginrestrictions() API.

– SU

Connection to Db2 is denied to users who are denied by AIX to become a substitute user with the su command. This option is equivalent to the S_SU mode of the loginrestrictions() API.

- LOCAL (or the variable is not set)

Connection to Db2 is denied to users who are denied AIX login. This option is equivalent to the S_LOGIN option of the loginrestrictions() API.

– NONE

The restrictions that affect the REMOTE, SU, or LOCAL options are not considered with the NONE option. This option is equivalent to the mode θ option of the loginrestrictions() API.

No matter which one of these options you set, user accounts or IDs that have the specified privileges are able to use Db2 successfully both locally on the server and from remote clients. For a description of the loginrestrictions() API, refer to AIX documentation.

DB2N0DE

- Operating system: All
- Default: NULL, Values: 1 to 999
- Used to specify the target logical node of a database partition server that you want to attach to or connect to. If this variable is not set, the target logical node defaults to the logical node which is defined with port 0 on the machine. In a partitioned database environment, the connection settings could have an impact on acquiring trusted connections. For example, if the **DB2NODE** variable is set to a node such that the establishment of a connection on that node requires going through an intermediate node (a hop node), it is the IP address of that intermediate node and the connection node that are considered when evaluating this connection in order to determine whether or not it can be marked as a trusted connection. In other words, it is not the original node from which the connection was initiated that is considered. Rather, it is the hop node that is considered.

Note: You cannot use the **db2set** command to update this registry variable. For more information, see "Setting environment variables outside the profile registries on Windows" on page 4.

DB20PTIONS

- Operating system: All
- Default: NULL
- Used to set the command line processor options.

DB2_PARALLEL_IO

- Operating system: All
- Default: NULL or * (in a Db2 pureScale environment) Values: *TablespaceID*:[*n*],... – a comma-separated list of defined table spaces (identified by their numeric table space ID). If the prefetch size of a table space is AUTOMATIC, you can indicate to the Db2 database manager the number of disks per container for that table space by specifying the table space ID, followed by a colon, followed by the number of disks per container, *n*. If *n* is not specified, the default is 6.

You can replace *TablespaceID* with an asterisk (*) to specify all table spaces. For example, if **DB2_PARALLEL_I0=***, all table spaces use six as the number of disks per container. If you specify both an asterisk (*) and a table space ID, the table space ID setting takes precedence. For example, if **DB2_PARALLEL_I0 =***,1:3, all table spaces use six as the number of disks per container, except for table space 1, which uses three.

This registry variable is used to change the way Db2 calculates the I/O parallelism of a table space. When I/O parallelism is enabled (either implicitly, by the use of multiple containers, or explicitly, by setting DB2_PARALLEL_IO), it is achieved by issuing the correct number of prefetch requests. Each prefetch request is a request for an extent of

pages. For example, a table space has two containers and the prefetch size is four times the extent size. If the registry variable is set, a prefetch request for this table space will be broken into four requests (one extent per request) with a possibility of four prefetchers servicing the requests in parallel.

You might want to set the registry variable if the individual containers in the table space are striped across multiple physical disks or if the container in a table space is created on a single RAID device that is composed of more than one physical disk.

If this registry variable is not set, the degree of parallelism of any table space is the number of containers of the table space. For example, if **DB2_PARALLEL_IO** is set to NULL and a table space has four containers, four extent-sized prefetch requests are issued; or if a table space has two containers and the prefetch size is four times the extent size, the prefetch request for this table space will be broken into two requests (each request will be for two extents).

If this registry variable is set, and the prefetch size of the table is not AUTOMATIC, the degree of parallelism of the table space is the prefetch size divided by the extent size. For example, if **DB2_PARALLEL_IO** is set for a table space that has a prefetch size of 160 and an extent size of 32 pages, five extent-sized prefetch requests are issued.

If this registry variable is set, and the prefetch size of the table space is AUTOMATIC, Db2 automatically calculates the prefetch size of a table space. The following table summarizes the different options available and how parallelism is calculated for each situation:

Prefetch size of table space	DB2_PARALLEL_I0 Setting	Parallelism is equal to:	
AUTOMATIC	Not set	Number of containers	
AUTOMATIC	Table space ID	Number of containers * 6	
AUTOMATIC	Table space ID:n	Number of containers * <i>n</i>	
Not AUTOMATIC	Not set	Number of containers	
Not AUTOMATIC	Table space ID	Prefetch size/extent size	
Not AUTOMATIC	Table space ID:n	Prefetch size/extent size	

Table 4. How Parallelism is Calculated

Disk contention might result using this variable in some scenarios. For example, if a table space has two containers and each of the two containers have each a single disk dedicated to it, setting the registry variable might result in contention on those disks because the two prefetchers will be accessing each of the two disks at once. However, if each of the two containers was striped across multiple disks, setting the registry variable would potentially allow access to four different disks at once.

To activate changes to this registry variable, issue a **db2stop** command and then enter a **db2start** command.

DB2PATH

• Operating system: Windows

- Default: Varies by operating system
- This environment variable is used to specify the directory where the product is installed on Windows 32-bit operating systems.

DB2_PMAP_COMPATIBILITY

- Operating system: All
- Default: OFF, Values: ON or OFF
- This variable allows users to continue using the sqlugtpi and sqlugrpn APIs to return, respectively, the distribution information for a table and the database partition number and database partition server number for a row. When this variable is set to 0N, the distribution map size for new databases is 4 096 entries. For the default value, or when this variable is set to 0FF, the distribution map size for new databases is increased to 32 768 entries. If you use the 32K distribution map, you need to use the new db2GetDistMap and db2GetRowPartNum APIs. This variable has no affect on existing databases.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2PROCESSORS

- Operating system: Windows
- Default: NULL, Values: 0–*n*-1 (where *n*= the number of processors)
- This variable sets the process affinity mask for a particular db2syscs process. In environments running multiple logical nodes, this variable is used to associate a logical node to a processor or set of processors.
 When specified, Db2 issues the SetProcessAffinityMask() api. If unspecified, the db2syscs process is associated with all processors on the server.

DB2RCMD_LEGACY_MODE

- Operating system: Windows,
- Default: NULL, Values: YES, ON, TRUE, or 1, or NO, OFF, FALSE, or 0
- This variable allows users to enable or disable the Db2 Remote Command Service's enhanced security. To run the Db2 Remote Command Service in a secure manner, set DB2RCMD_LEGACY_MODE to N0, 0FF, FALSE, 0, or NULL. To run in legacy mode (without enhanced security), set DB2RCMD_LEGACY_MODE to YES, 0N, TRUE, or 1. The secure mode is only available if your domain controller is running Windows 2000 or later.

Note: If **DB2RCMD_LEGACY_MODE** is set to YES, ON, TRUE, or 1, all requests sent to the Db2 Remote Command Service are processed under the context of the requestor. To facilitate this, you must allow either or both the machine and service logon account to impersonate the client by enabling the machine and service logon accounts at the domain controller.

Note: If **DB2RCMD_LEGACY_MODE** is set to N0, OFF, FALSE, or 0, you must have SYSADM authority in order to have the Db2 Remote Command Service execute commands on your behalf.

DB2RESILIENCE

• Operating system: All

- Default: ON, Values: ON (TRUE or 1), or OFF (FALSE or 0)
- This registry variable can be used to control whether physical read errors are tolerated, and activates extended trap recovery. The default behavior is to tolerate read errors and activate extended trap recovery. To revert to the behavior of previous releases and force the database manager to shut down the instance, set the registry variable to 0FF. This registry variable does not affect the existing storage key support.

DB2_RESTORE_GRANT_ADMIN_AUTHORITIES

- Operating system: All
- Default: OFF, Values: ON or OFF
- If **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** is set to ON, and you are restoring to a new or existing database, then you will be granted SECADM, DBADM, DATAACCESS, and ACCESSCTRL authorities.
- The following methods of restore are supported when DB2_RESTORE_GRANT_ADMIN_AUTHORITIES is set to ON:
 - Split mirror backups
 - ACS Snapshot backups
 - Online and offline database backups with the **RESTORE DATABASE** command

Note: Note that this variable has no effect on table space restores; no additional authorities will be granted to the user issuing the restore operation.

• If **DB2_WORKLOAD** is set to SAP, **DB2_RESTORE_GRANT_ADMIN_AUTHORITIES** will be set to ON.

DB2_SKIP_CHK_TMP

- Operating system: Linux and UNIX
- Default: Not set
- If DB2_SKIP_CHK_TMP is not set to TRUE or true, setting the variable DB2WORKDIR to /tmp will fail.

DB2_SYMPHONY_WLM

- Operating System: All
- Default: varies, Values: accepts several 'key=value' pairs separated by a ':' separator

Informs the Db2 instance that it is running in an integrated cluster which involves mixed technologies such as Platform Symphony and Db2. Valid 'key=value' pairs are:

EGO_PATH=<path_to_executable>

This specifies the path to where the Platform Symphony EGO commands are found. This 'key=value' pair must always be provided. An example path for an integrated environment is: /opt/ibm/biginsights/HAManager/data/1.2.7/linux2.6-glibc2.5-ppc64/bin.

CONSUMER=<consumer_name>

Specifies the consumer name Db2 should use when requesting slots from Platform Symphony. Default value is the BigSQL_<instance_name> consumer name.

EGO_RES_GROUP=<resource_group>

Specifies the Platform Symphony resource group that slots should be requested from. Default value is the ComputeHosts resource group.

KEEP_ALIVE=<value>

Specifies the time, in minutes, between successive keep-alive commands sent to Platform Symphony to ensure that Db2 slots continue to be marked as 'allocated'. Defaults to 5 minutes. In the event of an abnormal failure by Db2, the Platform Symphony slots allocated by Db2 will be automatically released in at most 2*KEEP_ALIVE minutes after the failure.

When the **DB2_SYMPHONY_WLM** registry variable is properly configured, each Db2 member at member start-up will request the appropriate number of slots (as determined by the percentage of each machine that Db2 has been configured to consume) from Platform Symphony. Each will send keep-alive messages to Platform Symphony for as long as the Db2 member is running.

At Db2 member stop time, the Db2 member will free all allocated slots back to Platform Symphony.

DB2SYSTEM

- Operating system: Windows and UNIX
- Default: NULL
- Specifies the name that is used by your users and database administrators to identify the Db2 database server system. If possible, this name should be unique within your network.

This name aids users in identifying the system that contains the database they wish to access. A value for **DB2SYSTEM** is set at installation time as follows:

- On Windows the setup program sets it equal to the computer name specified for the Windows system.
- On UNIX systems, it is set equal to the UNIX system's TCP/IP hostname.

DB2TMPDIR

- Operating system: Linux and UNIX
- Default: Not set
- This variable is used to specify a path for the temporary files directory. If the **DB2TMPDIR** variable is not set, the /tmp directory is used.
- The **DB2TMPDIR** variable cannot be used in a pureScale environments.
- The /tmp directory will still be required for runtime activities when the **DB2TMPDIR** variable is set. You cannot make the /tmp directory fully inaccessible after you have set the **DB2TMPDIR** variable.

DB2_UPDDBCFG_SINGLE_DBPARTITION

- Operating system: All
- Default: Not set, Values: 0/FALSE/NO, 1/TRUE/YES
- DB2_UPDDBCFG_SINGLE_DBPARTITION enables you to revert to the behavior of previous versions of Db2, in which updates to a database configuration apply only to the local database partition or the database partition that is set by the DB2NODE registry variable. This allows for

backward compatibility support for any existing command scripts or applications that require this behavior.

When set to 1, TRUE, or, YES, this registry variable allows you to specify that any updates and resets to your database affect only a specific partition. If the variable is not set (the default), updates or changes to a database configuration act across all database partitions, when you do not specify a partition clause.

Note: This variable does not apply to update or reset requests made by calling ADMIN_CMD routines.

DB2_USE_PAGE_CONTAINER_TAG

- Operating system: All
- Default:NULL, Values: ON, NULL
- By default, Db2 stores a container tag in the first extent of each DMS container, whether it is a file or a device. The container tag is the metadata for the container. Before Db2 Version 8.1, the container tag was stored in a single page, and it thus required less space in the container. To continue to store the container tag in a single page, set **DB2 USE PAGE CONTAINER_TAG** to ON.

However, if you set this registry variable to 0N when you use RAID devices for containers, I/O performance might degrade. Because for RAID devices you create table spaces with an extent size equal to or a multiple of the RAID stripe size, setting the **DB2_USE_PAGE_CONTAINER_TAG** to 0N causes the extents not to line up with the RAID stripes. As a result, an I/O request might need to access more physical disks than would be optimal. Users are strongly advised against enabling this registry variable unless you have very tight space constraints, or you require behavior consistent with pre-Version 8 databases.

To activate changes to this registry variable, issue a **db2stop** command and then enter a **db2start** command.

DB2WORKDIR

- Operating system: Linux and UNIX
- Default: Not set
- This variable is used to specify a path for a temporary working directory for Db2 Installation and Configuration.

If the DB2WORKDIR variable is not set, the temporary working directory depends on the type of installation:

- For root install: the root user's home directory
- For non-root install: instance user's home directory

For security reasons, do not set DB2WORKDIR to /tmp.

DB2_WORKLOAD

- Operating system: All
- Default: Not set, Values: 1C, ANALYTICS, CM, COGNOS_CS, FILENET_CM, INFOR_ERP_LN, MAXIMO, MDM, SAP, TPM, WAS, WC, or WP
- Each value for DB2_WORKLOAD represents a specific grouping of several registry variables with predefined settings. Only one value for DB2_WORKLOAD can be in effect at any given time.
- These are the valid values:
 - **1C** Use this setting when you want to configure a set of registry variables in your database for 1C applications.

ANALYTICS

Use this setting before creating the database to establish an optimal default configuration for analytic workloads. The ANALYTICS option ensures that configuration parameters are automatically set as follows unless you disable the Configuration Advisor:

- The **dft_table_org** (default table organization for user tables) database configuration parameter is set to COLUMN.
- The **dft_degree** (default degree) database configuration parameter is set to ANY.
- The dft_extent_sz (default extent size) database configuration parameter is set to 4.
- The catalogcache_sz (catalog cache) database configuration parameter is set to a value that is higher than the value for a non-analytics workload.
- The values of the sortheap (sort heap) and sheapthres_shr (sort heap threshold for shared sorts) database configuration parameters are calculated specifically for an analytics workload. These settings take into account the additional memory requirements for processing column-organized data.
- The util_heap_sz (utility heap size) database configuration parameter is set to a value that takes into account the additional memory that is required to load the data into column-organized tables.
- The **auto_reorg** (automatic reorganization) database configuration parameter is set to ON.

Tip: Running the AUTOCONFIGURE command against an existing database when DB2_WORKLOAD is set to ANALYTICS has the same result.

The following extra choices are made automatically:

- The default database page size for a newly created database is set to 32 KB.
- A larger database shared sort heap is allocated.
- Intraquery parallelism is enabled for any workload (including SYSDEFAULTUSERWORKLOAD) that inherits the intrapartition parallelism setting from the intra_parallel (enable intrapartition parallelism) database manager configuration parameter, even if intra_parallel is set to NO.
- Concurrency control is enabled on SYSDEFAULTMANAGEDSUBCLASS.
- Automatic table maintenance performs space reclamation for column-organized tables by default.
- Automatic registry settings:
 - DB2_ANTIJOIN=EXTEND
 - DB2_USE_ALTERNATE_PAGE_CLEANING=ON

The self-tuning memory manager (STMM) is not turned on by default in a partitioned database environment when DB2_WORKLOAD is set to ANALYTICS.

CM Use this setting when you want to configure a set of registry variables in your database for IBM Content Manager.

COGNOS_CS

Use this setting when you want to configure a set of registry variables in your database for Cognos[®] Content Server.

FILENET_CM

Use this setting when you want to configure a set of registry variables in your database for Filenet Content Manager.

INFOR_ERP_LN

Use this setting when you want to configure a set of registry variables in your database for Infor ERP Baan.

- **MAXIMO** Use this setting when you want to configure a set of registry variables in your database for Maximo[®].
- MDM Use this setting when you want to configure a set of registry variables in your database for Master Data Management.
- **SAP** Use this setting when want to configure a set of registry variables in your database for the SAP environment.

When you have set **DB2_WORKLOAD**=SAP, the user table space SYSTOOLSPACE and the user temporary table space SYSTOOLSTMPSPACE are not automatically created. These table spaces are used for tables created automatically by the following wizards, utilities, or functions:

- Automatic maintenance
- SYSINSTALLOBJECTS stored procedure, if the table space input parameter is not specified
- GET_DBSIZE_INFO stored procedure

Without the SYSTOOLSPACE and SYSTOOLSTMPSPACE table spaces, you cannot use these wizards, utilities, or functions.

To be able to use these wizards, utilities, or functions, do either of the following:

 Manually create the SYSTOOLSPACE table space to hold the objects that the tools need (in a partitioned database environment, create this table space on the catalog partition). For example:

CREATE REGULAR TABLESPACE SYSTOOLSPACE IN IBMCATGROUP MANAGED BY SYSTEM USING ('SYSTOOLSPACE')

 Specifying a valid table space, call the SYSINSTALLOBJECTS stored procedure to create the objects for the tools, and specify the identifier for the particular tool.
 SYSINSTALLOBJECTS will create a table space for you. If you do not want to use SYSTOOLSSPACE for the objects, specify a different user-defined table space.

After completing at least one of these choices, create the SYSTOOLSTMPSPACE temporary table space (also on the catalog partition, if you're working in a partitioned database environment). For example:

CREATE USER TEMPORARY TABLESPACE SYSTOOLSTMPSPACE IN IBMCATGROUP MANAGED BY SYSTEM USING ('SYSTOOLSTMPSPACE') Once the table space SYSTOOLSPACE and the temporary table space SYSTOOLSTMPSPACE are created, you can use the wizards, utilities, or functions mentioned earlier.

- **TPM** Use this setting when want to configure a set of registry variables in your database for the Tivoli[®] Provisioning Manager.
- **WAS** Use this setting when you want to configure a set of registry variables in your database for WebSphere[®] Application Server.
- **WC** Use this setting when you want to configure a set of registry variables in your database for WebSphere Commerce.
- **WP** Use this setting when you want to configure a set of registry variables in your database for WebSphere Portal.

Communications variables

You use communication variables to help control the flow of data on Db2 network connections and within the Db2 environment itself. You can set communication variables to control default behaviors such as TCP/IP settings and communication manager activity.

DB2_ALLOW_WLB_WITH_SEQUENCES

- Operating system: All
- Default=N0, Values: N0, YES
- This registry variable controls whether applications that access sequences are allowed to participate in workload balancing.

When **DB2_ALLOW_WLB_WITH_SEQUENCES** is set to N0, applications that reference PREVIOUS VALUE or NEXT VALUE in an SQL sequence statement are prevented from participating in workload balancing.

When **DB2_ALLOW_WLB_WITH_SEQUENCES** is set to YES, applications that reference PREVIOUS VALUE or NEXT VALUE in an SQL sequence statement are not prevented from participating in workload balancing. Applications must generate the next sequence value with the NEXT VALUE FOR *sequence* expression in a transaction before it references the previous value of the sequence using the PREVIOUS VALUE FOR *sequence* expression. Because the application can participate in workload balancing, each transaction might run in a new session or a different session. Accessing a previous sequence value in a transaction without first generating the next sequence value results in a SQL0845N error.

Sequence considerations in a Db2 pureScale environment:

In a Db2 pureScale environment, if you use the CACHE and NO ORDER options, multiple caches might be active simultaneously. Requests for the next value from different members might not result in the assignment of values in strict numeric order. For example, members DB2A and DB2B are using the same sequence. DB2A gets the cache values 1-20, while DB2B gets the cache values 21-40. If DB2A requested the next value first, then DB2B requested, and then DB2A requested again, the order of values assigned would be 1, 21, 2.

Using the ORDER or NO CACHE option ensures that the values assigned to a sequence shared by one or more applications across multiple members are in strict numeric order. If ORDER is specified, then NO CACHE is implied even if CACHE n is specified.

DB2CHECKCLIENTINTERVAL

• Operating system: All, server only

- Default=100, Values: A numeric value that is greater than or equal to zero.
- This variable specifies the frequency of TCP/IP client connection verifications during an active transaction. It permits early detection of client termination, instead of waiting until after the completion of the query. If this variable is set to 0, no verification is performed.

Lower values cause more frequent checks. As a guide, for low frequency, use 100; for medium frequency, use 50; for high frequency use 10. The value is measured in an internal Db2 metric. The values represent a linear scale, that is, increasing the value from 50 to 100 doubles the interval. Checking more frequently for client status while executing a database request lengthens the time taken to complete queries. If the Db2 workload is heavy (that is, it involves many internal requests), setting **DB2CHECKCLIENTINTERVAL** to a low value has a greater impact on performance than in a situation where the workload is light.

DB2COMM

- Operating system: All, server only
- Default=NULL, Values: NPIPE, TCPIP, SSL
- This variable specifies the communication managers that are started when the database manager is started. If this variable is not set, no Db2 communications managers are started at the server. You can set this variable to any combination of the values, separated by commas.

Note: The value NPIPE is valid only in Windows operating systems.

DB2FCMCOMM

- Operating system: All supported Db2 Enterprise Server Edition platforms
- Default=TCPIP4, Values: TCPIP4 or TCPIP6
- This variable specifies how the host names in the db2nodes.cfg file are resolved. All host names are resolved as IPv4 or IPv6. If an IP address instead of a host name is specified in db2nodes.cfg, the form of the IP determines if IPv4 or IPv6 is used. If **DB2FCMCOMM** is not set, its default setting of IPv4 means that only IPv4 hosts can be started.

Note: If the IP format resolved from the hostname specified in db2nodes.cfg, or the IP format directly specified in db2nodes.cfg does not match the setting of **DB2FCMCOMM**, **db2start** will fail.

DB2_FORCE_NLS_CACHE

- Operating system: AIX, HP_UX, Solaris
- Default=FALSE, Values: TRUE or FALSE
- This variable is used to eliminate the chance of lock contention in multi-threaded applications. When this registry variable is TRUE, the code page and territory code information is saved the first time a thread accesses it. From that point, the cached information is used for any other thread that requests this information. This eliminates lock contention and results in a performance benefit in certain situations. This setting should not be used if the application changes locale settings between connections. It is probably not needed in such a situation because multi-threaded applications typically do not change their locale settings because it is not *thread safe* to do so.

DB2_PMODEL_SETTINGS

- Operating system: All
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.
- This registry variable controls a set of parameters that allow you to modify the behavior of various aspects of the Db2 internal infrastructure. Separate parameters with a semicolon, as in the following example: db2set

DB2_PMODEL_SETTINGS=MLN_REMOTE_LISTENER:TRUE;ENHANCED_ROLLBACK:TRUE;SRVLST_EQUAL_WEIGHT:TRUE

• Parameters:

ENHANCED_ROLLBACK

- Default: FALSE
- Values: TRUE, FALSE
- Use the ENHANCED_ROLLBACK parameter to improve rollback behavior for units of work on Db2 servers in partitioned database environments. If you set this option to TRUE, rollback requests for units of work are sent only to logical database partitions that participated in the transaction.

MLN_REMOTE_LISTENER

- Default: FALSE
- Values: TRUE, FALSE
- Use the MLN_REMOTE_LISTENER parameter to start a TCP/IP listener on each logical database partition. If you set this option to TRUE, applications can connect directly to each logical database partition instead of routing requests through the database partition server that is assigned to logical port 0.

If you set this option to TRUE, ensure that the additional TCP/IP listeners do not use ports that are required by other services.

SRVLST_EQUAL_WEIGHT

- Default: FALSE
- Values: TRUE, FALSE
- Use the **SRVLST_EQUAL_WEIGHT** parameter if you want non-zero member weights in the server list to always be identical, thereby overriding the default behavior in which member weights are computed based on load. Member weights as contained in the server list are used by a remote client to distribute workload when workload balancing (WLB) is enabled.

If you set this option to TRUE, WLB at the client translates into even workload distribution among members with no regard for the member load.

DB2RSHCMD

- Operating system: UNIX, Linux
- Default=rsh (remsh on HP-UX), Values are a full path name to rsh, remsh, or ssh
- By default, Db2 database system uses rsh as the communication protocol when starting remote database partitions and with the **db2_all** script to run utilities and commands on all database partitions. For example, setting this registry variable to the full path name for ssh causes Db2 database products to use ssh as the communication protocol for the requested running of the utilities and commands. It may also be set to

the full path name of a script that invokes the remote command program with appropriate default parameters. This variable is only required for partitioned databases, or for single-partition environments where the **db2start** command is run from a different server than where the Db2 product was installed and for rsh or ssh dependant utilities that have the capability of starting, stopping or monitoring a Db2 instance, such as **db2gcf**. The instance owner must be able to use the specified remote shell program to log in from each Db2 database node to each other Db2 database node, without being prompted for any additional verification or authentication (that is, passwords or password phrases).

For detailed instructions on setting the **DB2RSHCMD** registry variable to use a ssh shell with Db2, see the white paper "Configure DB2 Universal DatabaseTM for UNIX to use OpenSSH."

DB2RSHTIMEOUT

- Operating system: UNIX, Linux
- Default=30 seconds, Values: 1 120
- This variable is only applicable if **DB2RSHCMD** is set to a non-null value. This registry variable is used to control the timeout period that the Db2 database system will wait for any remote command. After this timeout period, if no response is received, the assumption is made that the remote database partition is not reachable and the operation has failed.

Note: The time value given is not the time required to run the remote command, it is the time needed to authenticate the request.

DB2SORCVBUF

- Operating system: All
- Default=65 536
- Specifies the value of TCP/IP receive buffers.

DB2SOSNDBUF

- Operating system: All
- Default=65 536
- Specifies the value of TCP/IP send buffers.

DB2TCP_CLIENT_CONTIMEOUT

- Operating system: All, client only
- Default=0 (no timeout), Values: 0 32 767 seconds
- The DB2TCP_CLIENT_CONTIMEOUT registry variable specifies the number of seconds that a client must wait for the completion on a TCP/IP connect operation. A database connect operation via TCP/IP involves both connect() and recv() socket subroutines. The database manager returns the -30081 selectForConnectTimeout error if the connect() subroutine reaches the timeout, and the -30081 selectForRecvTimeout error if the recv() subroutine reaches the timeout.

There is no timeout if the registry variable is not set or is set to 0.

Note: Operating systems also have a connection timeout value that may take effect prior to the timeout you set using **DB2TCP_CLIENT_CONTIMEOUT**. For example, AIX has a default *tcp_keepinit*=150 (in half seconds) that would terminate the connection after 75 seconds.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2TCP_CLIENT_KEEPALIVE_TIMEOUT

- Operating system: AIX, HP-UX, Linux, Windows (client only)
- Default=15, Values: 0 32 767 seconds
- The **DB2TCP_CLIENT_KEEPALIVE_TIMEOUT** registry variable represents the total amount of time in seconds a socket can remain idle; and not respond to keepalive probes before it is considered unresponsive. When a socket is considered unresponsive, it is terminated by Db2. It is the client-side equivalent of **DB2TCP_SERVER_KEEPALIVE_TIMEOUT**.

When this variable is not set, the default setting of 15 seconds is used. When the **KeepAliveTimeout** keyword is set to 0, the keepalive setting that is set in the operating system takes effect. If set, this variable takes precedence over any keepAliveTimeout setting as specified in the db2dsdriver.cfg file.

Changes to this variable take effect immediately for all future TCP/IP connections and attachments to the server

DB2TCP_CLIENT_RCVTIMEOUT

- Operating system: All, client only
- Default=0 (no timeout), Values: 0 32 767 seconds
- The **DB2TCP_CLIENT_RCVTIMEOUT** registry variable specifies the number of seconds a client waits for data on a TCP/IP receive operation. If data from the server is not received in the seconds specified, then the Db2 database manager returns the error -30081 selectForRecvTimeout.

There is no timeout if the registry variable is not set or is set to 0.

Note: The value of the **DB2TCP_CLIENT_RCVTIMEOUT** can be overridden by the CLI, using the db2cli.ini keyword **ReceiveTimeout** or the connection attribute SQL_ATTR_RECEIVE_TIMEOUT.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2TCPCONNMGRS

- Operating system: All
- Default=1 on serial machines; square root of the number of processors rounded up to a maximum of sixteen connection managers on symmetric multiprocessor machines. Values: 1 to 16
- The default number of connection managers is created if the registry variable is not set. If the registry variable is set, the value assigned here overrides the default value. The number of TCP/IP connection managers specified up to a maximum of 16 is created. If less than 1 is specified then **DB2TCPCONNMGRS** is set to a value of 1 and a warning is logged that the value is out of range. If greater than 16 is specified then **DB2TCPCONNMGRS** is set to a value of 16 and a warning is logged that the value is out of range. Values between 1 and 16 are used as given. When there is greater than one connection manager created, connection throughput should improve when multiple client connections are received simultaneously. There may be additional TCP/IP connection manager processes (on UNIX) or threads (on Windows operating

systems) if the user is running on a SMP machine, or has modified the **DB2TCPCONNMGRS** registry variable. Additional processes or threads require additional storage.

Note: Having the number of connection managers set to 1 causes a drop in performance on remote connections in systems with a lot of users, frequent connects and disconnects, or both.

DB2TCP_SERVER_KEEPALIVE_TIMEOUT

- Operating system: AIX, HP-UX, Linux, Windows (server only)
- Default=60, Values: 0 32 767 seconds

•

The **DB2TCP_SERVER_KEEPALIVE_TIMEOUT** registry variable specifies the maximum time in seconds before an unresponsive TCP/IP client connection or attachment is detected as no longer alive. It is the server-side equivalent of **DB2TCP_CLIENT_KEEPALIVE_TIMEOUT** and keepAliveTimeout. When this variable is not set, the default setting of 60 seconds is used.

Changes to this variable take effect immediately for all future TCP/IP connections and attachments to the server. There is no need to restart the server instance.

Command line variables

You can set command line variables to control the default behavior of the command line in a Db2 product environment.

DB2BQTIME

- Operating system: All
- Default=1 second, Minimum value: 1 second
- This variable specifies the amount of time the command-line processor front end sleeps before it checks whether the back-end process is active and establishes a connection to it.

DB2BQTRY

- Operating system: All
- Default=60 retries, Minimum value: 0 retries
- This variable specifies the number of times the command-line processor front-end process tries to determine whether the back-end process is already active. It works in conjunction with DB2BQTIME.

DB2_CLP_EDITOR

- Operating system: All
- Default: Notepad(Windows), vi (UNIX), Values: Any valid editor that is located in the operating system path

Note: This registry variable is not set to the default value during installation. Instead, the code that makes use of this variable uses a default value if the registry variable is not set.

• This variable determines the editor to be used when executing the **EDIT** command. From a CLP interactive session, the **EDIT** command launches an editor preloaded with a user-specified command which can then be edited and run.

DB2_CLP_HISTSIZE

- Operating system: All
- Default: 20, Values: 1-500 inclusive

Note: This registry variable is not set to the default value during installation. Instead, the code that makes use of this variable uses a default value of 20 if the registry variable is not set or if it is set to a value outside of the valid range.

• This variable determines the number of commands stored in the command history during CLP interactive sessions. Because the command history is held in memory, a very high value for this variable might result in a performance impact depending on the number and length of commands run in a session.

DB2_CLPPROMPT

- Operating system: All
- Default=None (if it is not defined, "db2 => " will be used as the default CLP interactive prompt), Values: Any text string of length less than 100 that contains zero or more of the following tokens %i, %d, %ia, %da, or %n. Users need not set this variable unless they explicitly wish to change the default CLP interactive prompt (db2 =>).
- This registry variable allows a user to define the prompt to be used in the Command Line Processor (CLP) interactive mode. The variable can be set to any text string of length less than 100 characters containing zero or more of the optional tokens %i, %d, %ia, %da, or %n. When running in CLP interactive mode, the prompt to be used is constructed by taking the text-string specified in the DB2_CLPPROMPT registry variable and replacing all occurrences of the tokens %i, %d, %ia, %da, or %n by the local alias of the current attached instance, the local alias of the current attached instance, the local alias of the network the authorization ID of the current attached instance, the authorization and newline (that is, a carriage-return) respectively.

Note:

- 1. If the **DB2_CLPPROMPT** registry variable is changed within CLP interactive mode, the new value for **DB2_CLPPROMPT** will not take effect until the CLP interactive mode has been closed and reopened.
- 2. If no instance attachment exists, %ia is replaced by the empty string and %i is replaced by the value of the **DB2INSTANCE** registry variable. On Windows platforms only, if **DB2INSTANCE** is not set, %i is replaced by the value of the **DB2INSTDEF** registry variable. If neither of these variables are set, %i is replaced by the empty string.
- **3.** If no database connection exists, %da is replaced by the empty string and %d is replaced by the value of the **DB2DBDFT** registry variable. If the **DB2DBDFT** variable is not set, %d is replaced by the empty string.
- 4. The interactive input prompt will always present the values for the authorization IDs, database names, and instance names in upper case.

DB2IQTIME

- Operating system: All
- Default=5 seconds, Minimum value: 1 second
- This variable specifies the amount of time the command line processor back end process waits on the input queue for the front end process to pass commands.

DB2RQTIME

- Operating system: All
- Default=5 seconds, Minimum value: 1 second
- This variable specifies the amount of time the command line processor back end process waits for a request from the front end process.

Partitioned database environment variables

You use partitioned database environment variables to control the default behavior of a partitioned database environment, including authorization, failover, and network behavior.

DB2CHGPWD_EEE

- Operating system: Db2 ESE on AIX, Linux, and Windows
- Default=NULL, Values: YES or NO
- This variable specifies whether you allow other users to change passwords on AIX or Windows ESE systems. You must ensure that the passwords for all database partitions or nodes are maintained centrally using either a Windows domain controller on Windows, or LDAP on AIX. If not maintained centrally, passwords may not be consistent across all database partitions or nodes. This could result in a password being changed only at the database partition to which the user connects to make the change.

DB2_FCM_SETTINGS

- Operating system: Linux
- Default=YES, Values:
 - FCM_MAXIMIZE_SET_SIZE:[YES|TRUE|N0|FALSE]. The default value for FCM_MAXIMIZE_SET_SIZE is YES.
 - FCM_CFG_BASE_AS_FLOOR: [YES | TRUE | NO | FALSE]. The default value for FCM_CFG_BASE_AS_FLOOR is NO.
- You can set the **DB2_FCM_SETTINGS** registry variable with the FCM_MAXIMIZE_SET_SIZE token to preallocate a default 4 GB of space for the fast communication manager (FCM) buffer. The token must have a value of either YES or TRUE to enable this feature.

You can use the **DB2_FCM_SETTINGS** registry variable with the FCM_CFG_BASE_AS_FLOOR option to set the base value as the floor for the *fcm_num_buffers* and *fcm_num_channels* database manager configuration parameters. When the FCM_CFG_BASE_AS_FLOOR option is set to YES or TRUE, and these parameters are set to AUTOMATIC and have an initial or starting value, Db2 will not tune them below this value.

DB2_FORCE_OFFLINE_ADD_PARTITION

- Operating system: All
- Default=FALSE, Values: FALSE or TRUE
- This variable allows you to specify that add database partition server operations are to be performed offline. The default setting of FALSE indicates that Db2 database partition servers can be added without taking the database offline. However, if you want the operation to be performed offline or if some limitation prevents you from adding database partition servers when the database is online, set DB2_FORCE_OFFLINE_ADD_PARTITION to TRUE. When this variable is set to TRUE, new Db2 database partition servers are added according to the

Version 9.5 and earlier versions' behavior; that is, new database partition servers are not visible to the instance until it has been shut down and restarted.

DB2_NUM_FAILOVER_NODES

- Operating system: All
- Default=2, Values: 0 to the required number of database partitions
- Set **DB2_NUM_FAILOVER_NODES** to specify the number of additional database partitions that might need to be started on a machine in the event of failover.

In a Db2 database high availability solution, if a database server fails, the database partitions on the failed machine can be restarted on another machine. The fast communication manager (FCM) uses **DB2_NUM_FAILOVER_NODES** to calculate how much memory to reserve on each machine to facilitate this failover.

For example, consider the following configuration:

- Machine A has two database partitions: 1 and 2.
- Machine B has two database partitions: 3 and 4.
- DB2_NUM_FAILOVER_NODES is set to 2 on both A and B.

At START DBM, FCM will reserve enough memory on both A and B to manage up to four database partitions so that if one machine fails, the two database partitions on the failed machine can be restarted on the other machine. If machine A fails, database partitions 1 and 2 can be restarted on machine B. If machine B fails, database partitions 3 and 4 can be restarted on machine A.

DB2_PARTITIONEDLOAD_DEFAULT

- Operating system: All supported ESE platforms
- Default=YES, Values: YES or NO
- The DB2_PARTITIONEDLOAD_DEFAULT registry variable lets users change the default behavior of the load utility in an ESE environment when no ESE-specific load options are specified. The default value is YES, which specifies that in an ESE environment if you do not specify ESE-specific load options, loading is attempted on all database partitions on which the target table is defined. When the value is NO, loading is attempted only on the database partition to which the load utility is currently connected.

Note: This variable is deprecated and may be removed in a later release. The LOAD command has various options that can be used to achieve the same behavior. You can achieve the same results as the NO setting for this variable by specifying the following with the **LOAD** command: PARTITIONED DB CONFIG MODE LOAD_ONLY OUTPUT_DBPARTNUMS x, where x is the partition number of the partition into which you want to load data.

DB2PORTRANGE

- · Operating system: Windows
- Values: nnnn:nnnn
- This value is set to the TCP/IP port range used by FCM so that any additional database partitions created on another machine will also have the same port range.

DB2_DEFAULT_TABLE_DISTRIBUTION

- Operating system: All
- Default=NULL
- Values: RANDOM If explicit DISTRIBUTE BY clause is missing from **CREATE TABLE** statement, create table as DISTRIBUTE BY RANDOM. Setting is silently ignored for tables that can't be created as DISTRIBUTE BY RANDOM.

Db2 pureScale environment variables

These environment variables are specific to a Db2 pureScale environment: DB2_DATABASE_CF_MEMORY, DB2_MCR_RECOVERY_PARALLELISM_CAP, DB2_SD_ALLOW_SLOW_NETWORK.

DB2_DATABASE_CF_MEMORY

- Operating system: All
- Default: AUTO, Values: -1, or 0 to 100, or AUTO
- Type: Float
- When DB2_DATABASE_CF_MEMORY is unset or set to AUT0, CF memory is self-tuned for all databases that have these database configuration parameters set to AUT0MATIC: cf_gbp_sz, cf_lock_sz, and cf_sca_sz. DB2_DATABASE_CF_MEMORY can be changed dynamically, and a database manager restart is not required.
- DB2_DATABASE_CF_MEMORY is used to indicate the proportion of total CF memory (CF_MEM_SZ) assigned to each database that has database configuration parameter cf_db_mem_sz set to AUTOMATIC. Any database that has cf_db_mem_sz set to a specific value ignores this registry variable.
- For all databases to have an equal share of the CF memory resources when more that 100 databases are active, the **DB2_DATABASE_CF_MEMORY** registry variable must use values less than 1. For example, if there are 200 active databases, each with an equal share of the CF memory, this registry variable must be set to 0.5.
- Use of the DB2_DATABASE_CF_MEMORY registry variable must be coordinated with the cf_db_mem_sz and numdb configuration parameters. For more information, see Db2 pureScale CF memory parameter configuration.

DB2_MCR_RECOVERY_PARALLELISM_CAP

- Operating system: All
- In a multiple database environment, if member crash recovery (MCR) is required, the number of databases recovered in parallel on each member is set by the value of the **numdb** configuration parameter or the **DB2_MCR_RECOVERY_PARALLELISM_CAP** registry variable, whichever value is smaller.

DB2_SD_ALLOW_SLOW_NETWORK

- Operating system: All
- Default: OFF, Values: ON, OFF
- This variable allows an Ethernet card that is less than 10GE to use the TCP option of the **cf_transport_method** configuration parameter.
- This lower transfer rate Ethernet card limits performance.

Query compiler variables

You can set query compiler variables to control optimization decisions about your database, such as forcing optimization decisions and SQL query operations.

DB2_ANTIJOIN

- Operating system: All
- Default=NO in an ESE environment, Default=EXTEND in a non-ESE environment, Values: YES, NO, or EXTEND
- If DB2_WORKLOAD is set to ANALYTICS, DB2_ANTIJOIN is set to EXTEND.
- When this variable is set to YES, the optimizer searches for opportunities to transform NOT EXISTS subqueries into anti-joins which can be processed more efficiently by Db2.

When this variable is set to N0, the optimizer limits the opportunities to transform NOT EXISTS subqueries into anti-joins.

When this variable is set to EXTEND, the optimizer searches for opportunities to transform both NOT IN and NOT EXISTS subqueries into anti-joins.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_DEFERRED_PREPARE_SEMANTICS

- Operating system: All
- Default=NO, Values: YES or NO
- When set to YES, this registry variable enables deferred prepare semantics such that all untyped parameter markers used in PREPARE statements will derive their data types and length attributes based on the input descriptor associated with the subsequent OPEN or EXECUTE statements. This allows untyped parameter markers to be used in more places than was supported previously.

Note: Setting **DB2_DEFERRED_PREPARE_SEMANTICS** to YES may cause unintended effects or results. In cases where the data type in the input descriptor is different from the data type derived using the rules for "Determining data types of untyped expressions," the following can occur:

- The query performance is degraded because of the additional cast operation.
- The query fails because a data type cannot be converted.
- The query can return different results.

For example, assume a table t1, with a column char_col which is defined as VARCHAR(10) with values '1', '100', '200', 'xxx'. A user runs the following query:

select * from t1 where char_col = ?

If the data type of the input parameter is INTEGER, and deferred prepare is being used, the column char_col is cast to numeric. However, the query fails because one of the rows in the table contains non-numeric data ('xxx') which cannot be converted to a numeric value. When set to YES_DBCS_GRAPHIC_TO_CHAR, this registry variable specifies that parameter markers are to be typed as VARCHAR instead of

VARGRAPHIC. The **DB2_DEFERRED_PREPARE_SEMANTICS** registry variable has this setting implicitly if all of the following are true:

- **DB2_DEFERRED_PREPARE_SEMANTICS** is not set (that is, set to NULL).

- The DB2_COMPATIBILITY_VECTOR registry variable is set to ORA, MYS, or MSS.
- You are in a double-byte character set (DBCS) environment.

The **DB2_DEFERRED_PREPARE_SEMANTICS** registry variable must be set prior to issuing the **db2start** command.

This registry variable is only recommended for Unicode and SBCS databases.

DB2_INLIST_TO_NLJN

- Operating system: All
- Default=NO, Values: YES or NO
- In some situations, the SQL and XQuery compiler can rewrite an IN list predicate to a join. For example, the following query:

```
SELECT *
FROM EMPLOYEE
WHERE DEPTNO IN ('D11', 'D21', 'E21')
```

could be written as:

```
SELECT *
FROM EMPLOYEE, (VALUES 'D11', 'D21', 'E21) AS V(DNO)
WHERE DEPTNO = V.DNO
```

This revision might provide better performance if there is an index on DEPTNO. The list of values would be accessed first and joined to EMPLOYEE with a nested loop join using the index to apply the join predicate.

Sometimes the optimizer does not have accurate information to determine the best join method for the rewritten version of the query. This can occur if the IN list contains parameter markers or host variables which prevent the optimizer from using catalog statistics to determine the selectivity. This registry variable causes the optimizer to favor nested loop joins to join the list of values, using the table that contributes the IN list as the inner table in the join.

Note: When either or both of the Db2 query compiler variables **DB2_MINIMIZE_LISTPREFETCH** and **DB2_INLIST_T0_NLJN**, are set to YES, they remain active even if REOPT(ONCE) is specified.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_LIKE_VARCHAR

- Operating system: All
- Default=Y,Y,
- Controls the use of sub-element statistics. These are statistics about the content of data in columns when the data has a structure in the form of a series of sub-fields or sub-elements delimited by blanks. Collection of sub-element statistics is optional and controlled by options in the **RUNSTATS** command or API.

Important: This variable is deprecated and might be removed in a future release because you should only change the settings under the advisement of IBM service.

This registry variable affects how the optimizer deals with a predicate of the form:

COLUMN LIKE '%xxxxx%'

where the xxxxx is any string of characters.

The syntax showing how this registry variable is used is:

```
db2set DB2_LIKE_VARCHAR=[Y|N|S|num1] [,Y|N|S|num2]
```

where

- The term preceding the comma, or the only term to the right of the predicate, means the following but only if the second term is specified as N or the column does not have positive sub-element statistics:
 - S The optimizer estimates the length of each element in a series of elements concatenated together to form a column based on the length of the string enclosed in the % characters.
 - Y The default. Use a default value of 1.9 for the algorithm parameter. Use a variable-length sub-element algorithm with the algorithm parameter.
 - N Use a fixed-length sub-element algorithm.
 - num1 Use the value of num1 as the algorithm parameter with the variable length sub-element algorithm.
- The term following the comma means the following, but only for columns that do have positive sub-element statistics:
 - N Do not use sub-element statistics. The first term takes effect
 - Y The default. Use a variable-length sub-element algorithm that uses sub-element statistics together with the 1.9 default value for the algorithm parameter in the case of columns with positive sub-element statistics.
 - num2 Use a variable-length sub-element algorithm that uses sub-element statistics together with the value of num2 as the algorithm parameter in the case of columns with positive sub-element statistics.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_MINIMIZE_LISTPREFETCH

- Operating system: All
- Default=NO, Values: YES or NO
- List prefetch is a special table access method that involves retrieving the qualifying RIDs from the index, sorting them by page number and then prefetching the data pages. Sometimes the optimizer does not have accurate information to determine if list prefetch is a good access method. This might occur when predicate selectivities contain parameter markers or host variables that prevent the optimizer from using catalog statistics to determine the selectivity.

This registry variable prevents the optimizer from considering list prefetch in such situations.

Note: When either or both of the Db2 query compiler variables **DB2_MINIMIZE_LISTPREFETCH** and **DB2_INLIST_T0_NLJN**, are set to YES, they remain active even if REOPT(ONCE) is specified.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_NEW_CORR_SQ_FF

- Operating system: All
- Default=OFF, Values: ON or OFF
- Affects the selectivity value computed by the query optimizer for certain subquery predicates when it is set to 0N. It can be used to improve the accuracy of the selectivity value of equality subquery predicates that use the MIN or MAX aggregate function in the SELECT list of the subquery. For example:

SELECT * FROM T WHERE T.COL = (SELECT MIN(T.COL) FROM T WHERE ...)

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_OPT_MAX_TEMP_SIZE

- Operating system: All
- Default=NULL, Values: amount of space in megabytes that can be used by a query in all temporary table spaces
- Limits the amount of space that queries can use in the temporary table spaces. Setting DB2_OPT_MAX_TEMP_SIZE can cause the optimizer to choose a plan that is more expensive than would otherwise be chosen, but which uses less space in the temporary table spaces. If you set DB2_OPT_MAX_TEMP_SIZE, be sure to balance your need to limit use of temporary table space against the efficiency of the plan your setting causes to be chosen.

If **DB2_WORKLOAD=SAP** is set, **DB2_OPT_MAX_TEMP_SIZE** is automatically set to 10 240 (10 GB).

If you run a query that uses temporary table space in excess of the value set for **DB2_OPT_MAX_TEMP_SIZE**, the query does not fail, but you receive a warning that its performance may be suboptimal, as not all resources may be available.

The operations considered by the optimizer that are affected by the limit set by **DB2_OPT_MAX_TEMP_SIZE** are:

- Explicit sorts for operations such as ORDER BY, DISTINCT, GROUP BY, merge scan joins, and nested loop joins.
- Explicit temporary tables
- Implicit temporary tables for hash joins and duplicate merge joins

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_REDUCED_OPTIMIZATION

- Operating system: All
- Default=NO, Values: NO, YES, any integer, DISABLE, JUMPSCAN, NO_SORT_NLJOIN, or NO_SORT_MGJOIN
- This registry variable lets you request either reduced optimization features or rigid use of optimization features at the specified

optimization level. If you reduce the number of optimization techniques that are used, you also reduce time and resource use during optimization.

If you set this variable, the following syntax rules apply:

- Separate each option with a comma (,), and ensure that no spaces appear before or after the comma.
- Separate an option and the value for that option with a single space.
- If the setting includes a space, enclose the setting in double quotation marks ("").

The following example shows the correct syntax: db2set DB2_REDUCED_OPTIMIZATION="NO_SORT_NLJOIN,JUMPSCAN ON"

Note: Although optimization time and resource use might be reduced, the risk of producing a less than optimal data access plan is increased. Use this registry variable only when advised by IBM or one of its partners.

– If set to NO

The optimizer does not change its optimization techniques.

If set to YES

If the optimization level is 5 (the default) or lower, the optimizer disables some optimization techniques that might consume significant prepare time and resources but do not usually produce a better access plan.

If the optimization level is exactly 5, the optimizer scales back or disables some additional techniques, which might further reduce optimization time and resource use, but also further increase the risk of a less than optimal access plan. For optimization levels lower than 5, some of these techniques might not be in effect in any case. If they are, however, they remain in effect.

If set to any integer

The effect is the same as YES, with the following additional behavior for dynamically prepared queries optimized at level 5. If the total number of joins in any query block exceeds the setting, then the optimizer switches to greedy join enumeration instead of disabling additional optimization techniques as described previously for level 5 optimization levels. which implies that the query will be optimized at a level similar to optimization level 2.

- If set to DISABLE

The behavior of the optimizer when unconstrained by this **DB2_REDUCED_OPTIMIZATION** variable is sometimes to dynamically reduce the optimization for dynamic queries at optimization level 5. This setting disables this behavior and requires the optimizer to perform full level 5 optimization.

– If set to JUMPSCAN

Use this option to control if the Db2 optimizer can use jump scan operations. You can specify the following values:

- OFF = The Db2 optimizer will not create plans using jumps cans.
- ON = The Db2 optimizer uses cost-based analysis to determine whether to generate plans that use jump scans (default).
- If set to NO_SORT_NLJOIN

The optimizer does not generate query plans that force sorts for nested loop joins (NLJN). These types of sorts can be useful for improving performance; therefore, be careful when using the NO_SORT_NLJOIN option, as performance can be severely impacted.

- If set to NO_SORT_MGJOIN

The optimizer does not generate query plans that force sorts for merge scan joins (MSJN). These types of sorts can be useful for improving performance; therefore, be careful when using the NO SORT MGJOIN option, as performance can be severely impacted.

Note that the dynamic optimization reduction at optimization level 5 takes precedence over the behavior described for optimization level of exactly 5 when **DB2_REDUCED_OPTIMIZATION** is set to YES as well as the behavior described for the integer setting.

– If set to ZZJN:

Use this option to control how the Db2 optimizer uses the zigzag join method for star schema-based queries that contain one fact table. You can specify the following values:

- OFF = The Db2 optimizer does not use the zigzag join method.
- ON = The Db2 optimizer uses cost-based analysis to determine whether to use the zigzag join method or a different join method (default).
- FORCE = If the zigzag join method is feasible, the Db2 optimizer uses the zigzag join method.
- If set to ZZJN_MULTI_FACT:

Use this option to control how the Db2 optimizer uses the zigzag join method for star schema-based queries that contain more than one fact table. You can specify the following values:

- OFF = The Db2 optimizer does not use the zigzag join method.
- ON = The Db2 optimizer uses cost-based analysis to determine whether to use the zigzag join method or a different join method (default).
- FORCE = If the zigzag join method is feasible, the Db2 optimizer uses the zigzag join method.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_SELECTIVITY

- Operating system: All
- Default=NO, Values: YES or NO
- This registry variable controls where the SELECTIVITY clause can be used in search conditions in SQL statements.

When this registry variable is set to N0, the SELECTIVITY clause can only be specified in a user-defined predicate.

When this registry variable is set to YES, the SELECTIVITY clause can be specified for the following predicates:

- A user-defined predicate
- A basic predicate in which at least one expression contains host variables or parameter markers

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_SQLROUTINE_PREPOPTS

- Operating system: All
- Default=Empty string, Values:
 - APREUSE {YES | NO}
 - BLOCKING {UNAMBIG | ALL | NO}
 - CONCURRENTACCESSRESOLUTION { USE CURRENTLY COMMITTED | WAIT FOR OUTCOME }
 - DATETIME {DEF | USA | EUR | ISO | JIS | LOC}
 - DEGREE {1 | *degree-of-parallelism* | ANY}
 - DYNAMICRULES {BIND | INVOKEBIND | DEFINEBIND | RUN | INVOKERUN | DEFINERUN}
 - EXPLAIN {NO | YES | ALL}
 - EXPLSNAP {NO | YES | ALL}
 - FEDERATED {NO | YES}
 - INSERT {DEF | BUF}
 - ISOLATION {CS | RR | UR | RS | NC}
 - OPTPROFILE {profile_name | schema_name.profile_name}
 - QUERYOPT optimization-level
 - REOPT {NONE | ONCE | ALWAYS}
 - STATICREADONLY {YES | NO | INSENSITIVE}
 - VALIDATE {RUN | BIND}
- The **DB2_SQLROUTINE_PREPOPTS** registry variable can be used to customize the precompile and bind options for SQL and XQuery procedures and functions. When setting this variable, separate each of the options with a space, as follows:

db2set DB2_SQLROUTINE_PREPOPTS="BLOCKING ALL VALIDATE RUN"

For a complete description of each option and its settings, see "BIND command."

If you want to achieve the same results as **DB2_SQLROUTINE_PREPOPTS** for select individual procedures, but without restarting the instance, use the SET_ROUTINE_OPTS procedure.

Performance variables

You can set performance variables to improve database processes, such as access plan optimizations, memory tuning operations, and operating resource policies.

DB2_ALLOCATION_SIZE

- Operating system: All
- Default=128 KB, Range: 64 KB 256 MB
- Specifies the size of memory allocations for buffer pools.

The potential advantage of setting a higher value for this registry variable is fewer allocations will be required to reach a desired amount of memory for a buffer pool.

The potential cost of setting a higher value for this registry variable is wasted memory if the buffer pool is altered by a non-multiple of the allocation size. For example, if the value for **DB2_ALLOCATION_SIZE** is 8 MB and a buffer pool is reduced by 4 MB, this 4 MB will be wasted because an entire 8 MB segment cannot be freed.

Note: DB2_ALLOCATION_SIZE is deprecated and may be removed in a later release.

DB2_APM_PERFORMANCE

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- Set this variable to 0N to enable performance-related changes in the access plan manager (APM) that affect the behavior of the query cache (package cache). These settings are not usually recommended for production systems. They introduce some limitations, such as the possibility of out-of-package cache errors or increased memory use, or both.

Setting **DB2_APM_PERFORMANCE** to ON also enables the NO PACKAGE LOCK mode. This mode allows the global query cache to operate without the use of package locks, which are internal system locks that protect cached package entries from being removed. The NO PACKAGE LOCK mode might result in somewhat improved performance, but certain database operations are not allowed. These prohibited operations might include: operations that invalidate packages, operations that inoperate packages, and **PRECOMPILE**, **BIND**, and **REBIND**.

DB2ASSUMEUPDATE

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- When enabled, this variable allows the Db2 database system to assume that all fixed-length columns provided in an UPDATE statement are being changed. This eliminates the need for the Db2 database system to compare the existing column values to the new values to determine if the column is actually changing. Using this registry variable can cause additional logging and index maintenance when columns are provided for update (for example, in a SET clause) but are not actually being modified.

The activation of the **DB2ASSUMEUPDATE** registry variable is effective on the **db2start** command.

DB2_AVOID_LOCK_ESCALATION

- Operating system: All
- Default=0FF (ON when DB2_WORKLOAD=SAP), Values: ON or 0FF
- When the **DB2_AVOID_LOCK_ESCALATION** registry variable is ON, lock escalation will not be performed. Instead, SQL0912N is returned to the application that requested the lock that would normally result in lock escalation. The application is able to either **COMMIT** or **ROLLBACK** which will free the locks held by this application. This variable can be updated online without restarting the instance.
- This variable is available in Version 11.1.2.2 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

DB2_AVOID_PREFETCH

• Operating system: All

- Default=0FF, Values: 0N or 0FF
- Specifies whether prefetch should be used during crash recovery. If DB2_AVOID_PREFETCH =0N, prefetch is not used.

DB2_BACKUP_USE_DI0

- Operating system: All
- Default: OFF, Values: ON or OFF
- Specifies whether or not backup and load copy images are cached by the operating system. The default behavior is to cache the image file. When DB2_BACKUP_USE_DIO is set to ON, the backup or load copy image file is directly written to disk, bypassing the file cache.

Setting this variable to 0N might result in the operating system better utilizing memory resources because there is no benefit to caching the backup or load copy image file. This performance impact will have the largest benefit for Linux platforms. However, there may be a slight slowdown of the backup or load itself, so you should measure the change in backup or load performance when DB2_BACKUP_USE_DIO is set to 0N.

Note: Changing the value of this registry variable does not affect the behavior of the backup or load that is already running. Changing the value will take effect when the next backup or load is run, and it does not require an instance restart.

DB2BPVARS

- Operating system: As specified for each parameter
- Default=Path
- Two sets of parameters are available to tune buffer pools. One set of parameters, available only on Windows, specify that buffer pools should use scatter read for specific types of containers. The other set of parameters, available on all platforms, affect prefetching behavior.

Important: This performance variable has been deprecated in Version 9.5 and might be removed in a future release. For more information, see .

Parameters are specified in an ASCII file, one parameter on each line, in the form parameter=value. For example, a file named bpvars.vars might contain the following line:

NO_NT_SCATTER = 1

Assuming that bpvars.vars is stored in F:\vars\, to set these variables you execute the following command:

db2set DB2BPVARS=F:\vars\bpvars.vars

Scatter-read parameters

The scatter-read parameters are recommended for systems with a large amount of sequential prefetching against the respective type of containers and for which you have already set **DB2NTNOCACHE** to ON. These parameters, available only on Windows platforms, are NT_SCATTER_DMSFILE, NT_SCATTER_DMSDEVICE, and NT_SCATTER_SMS. Specify theNO_NT_SCATTER parameter to explicitly disallow scatter read for any container. Specific parameters are used to turn scatter read on for all containers of the indicated type. For each of these parameters, the default is zero (or OFF); and the possible values include: zero (or OFF) and 1 (or ON). **Note:** You can turn on scatter read only if **DB2NTNOCACHE** is set to 0N to turn Windows file caching off. If **DB2NTNOCACHE** is set to 0FF or not set, a warning message is written to the administration notification log if you attempt to turn on scatter read for any container, and scatter read remains disabled.

DB2CHKPTR

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- Specifies whether or not pointer checking for input is required.

DB2CHKSQLDA

- Operating system: All
- Default=ON, Values: ON or OFF
- Specifies whether or not SQLDA checking for input is required.

DB2_EVALUNCOMMITTED

- Operating system: All
- Default: NO, Values: YES, NO
- When enabled, this variable allows, where possible, scans to defer or avoid row locking until the data is known to satisfy predicate evaluation. With this variable enabled, predicate evaluation may occur on uncommitted data. Only scans that are not Currently Committed (CC) applicable considers these variables.

DB2_EVALUNCOMMITTED is only applicable when currently committed semantics will not help avoid lock contentions. When this variable is set and currently committed is applicable to a scan, deleted rows will not be skipped and predicate evaluate will not occur on uncommitted data; the currently committed version of the rows and data will be processed instead.

As well, **DB2_EVALUNCOMMITTED** is applicable only to statements using either Cursor Stability or Read Stability isolation levels. Furthermore, deleted rows are skipped unconditionally on table scan access while deleted keys are not skipped for index scans unless the registry variable **DB2_SKIPDELETED** is also set.

The activation of the **DB2_EVALUNCOMMITTED** registry variable is effective on the **db2start** command. The decision as to whether deferred locking is applicable is made at statement compile or bind time.

DB2_EXTEND_COL_UNIQUE_INDEX_ACCESS

- Operating system: All
- Default=0FF, Values: 0N, 0FF
- This variable lets you control whether a modification state index is created when a unique or primary key constraint is created on a column-organized table. When this registry variable is set to ON, the modification state index, if it does not already exist, is generated when creating an enforced primary key or unique constraint.
 DB2_EXTEND_COL_UNIQUE_INDEX_ACCESS is automatically set to 'ON' when the DB2_WORKLOAD registry variable is set to 'SAP'. The modification state index is needed for certain index access plans to be chosen.

DB2_EXTENDED_I0_FEATURES

• Operating system: AIX

- Default=0FF, Values: 0N, 0FF
- Set this variable to 0N to enable features that enhance I/O performance. This enhancement includes improving the hit rate of memory caches as well as reducing the latency on high priority I/O. These features are only available on certain combinations of software and hardware configuration; setting this variable to 0N for other configurations will be ignored by either the Db2 database management system or by the operating system. The minimum configuration requirements are:
 - Database version: Db2 V9.1
 - RAW device must be used for database containers (container on file systems is not supported)
 - Storage subsystem: Shark DS8000[®] supports all the enhanced I/O performance features. Refer to the Shark DS8000 documentation for setup and prerequisite information.

The default I/O priority settings for HIGH, MEDIUM, and LOW are 3, 8, and 12, respectively; you can use the DB2_IO_PRIORITY_SETTING registry variable to change these settings.

DB2_EXTENDED_OPTIMIZATION

- Operating system: All
- Default: OFF, Values: ON, OFF, ENHANCED_MULTIPLE_DISTINCT, IXOR, or OPT_SORTHEAP_EXCEPT_COL value
- This variable specifies whether the query optimizer uses optimization extensions to help improve query performance. The values specify different optimization extensions. To specify multiple values, use a comma-separated list.

The default behavior (specified by the 0FF or IXOR value) is for the optimizer to extend the index ORing data access method to include OR predicates that reference any indexed column even when non-indexed column predicates are present. For example, consider the following two index definitions:

INDEX IX2: dept ASC INDEX IX3: job ASC

The following predicates can be satisfied by using these two indexes when the IXOR option is set:

WHERE

```
dept = :hv1 OR
(job = :hv2 AND
years >= :hv3)
```

You can use the OPT_SORTHEAP_EXCEPT_COL *value* option to override the value of the **sortheap** database configuration parameter. The override value affects query optimization only and does not determine the amount of actual memory that is available at run time. If the query accesses a column-organized table, this override value is ignored to allow the query compiler to use the current value of the **sortheap** database configuration parameter.

One usage of the OPT_SORTHEAP_EXCEPT_COL is for shadow tables. Shadow tables facilitate BLU Acceleration for analytical queries in OLTP environment. Shadow tables are column-organized tables. The requirements for sort heap memory are higher than you would normally have for databases in OLTP environments. To increase the sort heap memory without affecting existing access plans for OLTP queries, add

OPT_SORTHEAP_EXCEPT_COL to DB2_EXTENDED_OPTIMIZATION to override the value of the **sortheap** database configuration parameter.

DB2_EXTENDED_OPTIMIZATION settings might not improve query performance in all environments. You should test to determine individual query performance improvements.

Important:

- The ENHANCED_MULTIPLE_DISTINCT and IXOR values are deprecated as of Version 10.1 and might be removed in a future release. Removing the ENHANCED_MULTIPLE_DISTINCT option makes new enhancements that improve the performance of multiple distinct queries available. The IXOR value is redundant because it specifies the default behavior. For more details, see "Registry variables with changed behaviors" in *What's New for Db2 Version 10.5*.
- The ENHANCED_MULTIPLE_DISTINCT value takes effect dynamically only if it was enabled when the instance was last started.

DB2_IO_PRIORITY_SETTING

- Operating system: AIX
- Values: HIGH:#,MEDIUM:#,LOW:#, where # can be 1 to 15
- This variable is used in combination with the DB2_EXTENDED_IO_FEATURES registry variable. This registry variable provides a means to override the default HIGH, MEDIUM, and LOW I/O priority settings for the Db2 database system, which are 3, 8, and 12, respectively. This registry variable must be set prior to the start of an instance; any modification requires an instance restart. Note that setting this registry variable alone does not enable the enhanced I/O features, DB2_EXTENDED_IO_FEATURES must be set to enable them. All system requirements for DB2_EXTENDED_IO_FEATURES also apply to this registry variable.

DB2_KEEP_AS_AND_DMS_CONTAINERS_OPEN

- Operating system: All
- Default: NO, Values: YES or NO
- When you set this variable to 0N, each DMS table space container has a file handle opened until the database is deactivated. Query performance might improve because the overhead to open the containers is eliminated. You should use this registry only in pure DMS environments, otherwise performance of queries against SMS table spaces might be impacted negatively.

DB2_KEEPTABLELOCK

- Operating system: All
- Default: OFF, Values: ON, TRANSACTION, OFF, CONNECTION
- When this variable is set to 0N or TRANSACTION, this variable allows the Db2 database system to maintain the table lock when an Uncommitted Read or Cursor Stability isolation level is closed. The table lock that is kept is released at the end of the transaction, just as it would be released for Read Stability and Repeatable Read scans.

When this variable is set to CONNECTION, a table lock is released for an application until the application either rolls back the transaction or the connection is reset. The table lock continues to be held across commits and application requests to drop the table lock are ignored by the database. The table lock remains allocated to the application. Thus, when the application re-requests the table lock, the lock is already available.

For application workloads that can leverage this optimization, performance should improve. However, the workloads of other application executing concurrently might be impacted. Other applications might get blocked from accessing a given table resulting in poor concurrency. Db2 SQL catalog tables are not impacted by this setting. The CONNECTION setting also includes the behavior described with the ON or TRANSACTION setting.

This registry variable is checked at statement compile or bind time.

DB2_LARGE_PAGE_MEM

- Operating system: AIX, Linux, Windows Server
- Default: NULL, Values: Use * to denote all applicable memory regions that should use large page memory, or a comma-separated list of specific memory regions that should use large page memory. Available regions vary by operating system.
- Db2 uses the term "large pages" generically, whereas operating systems use the terms "large" and "huge" to reference both specific and various larger hardware-dependent page sizes. A "large" or "huge" OS page size is not the same size on all OS/hardware platforms, and these terms may refer to multiple page sizes on a given OS/hardware platform.
- The DB setting applies to the **DATABASE_MEMORY** region and enables the following page sizes: AIX large (16MB) pages, Linux x64 huge (2MB) pages, and Windows large (2MB) pages.- The DB:16GB setting is available on AIX only, which enables huge (16GB) pages for the **DATABASE_MEMORY** region.
- The PRIVATE, DBMS, FCM, and APPL (**APPL_MEMORY**) settings are available on AIX only, each of which enables large (16MB) pages for the applicable memory region.
- The * setting enables OS large/huge page memory for all of the available regions as indicated above. On AIX, large (16MB) pages will be enabled for database memory as opposed to huge (16GB) pages.

When you are using large or huge pages for database memory (DB), dynamic decreasing of database memory is limited, and the **db_mem_thresh** setting is ignored. The self-tuning memory manager (STMM) does not adjust the overall size of the database area. However, STMM tunes the areas inside the database memory region subject to configuration. For more information, see the **database_memory** topic in the related links section.

Memory access-intensive applications that use large amounts of virtual memory might obtain performance improvements by using large or huge pages. To enable theDb2 database system to use them, you must first configure the operating system to use large or huge pages.

To enable large pages for agent private memory on 64-bit Db2 for AIX (the DB2_LARGE_PAGE_MEM=PRIVATE setting), you must configure large pages on the operating system and the instance owner must possess the CAP_BYPASS_RAC_VMM and CAP_PROPAGATE capabilities.

On Linux, to verify that huge kernel pages are available, issue the following command:

cat /proc/meminfo

If huge kernel pages are available, the following three lines should be displayed (with different numbers depending on the amount of memory configured on your server):

HugePages Total:	200	
HugePages_Free:	200	
Hugepagesize:	16384	kВ

If you do not see these lines, or if the HugePages_Total is 0, you must configure the operating system or kernel.

On Windows, the amount of large page memory that is available on the system is less than the total available memory. After the system has been running for some time, memory can become fragmented, and the amount of large page memory decreases. The **DB2_ALLOCATION_SIZE** registry variable should be set to a high value, such as 256 MB, in order to achieve consistent performance allocating large memory pages on Windows. (Note that **DB2_ALLOCATION_SIZE** requires you to stop and restart the instance.)

DB2_LOGGER_NON_BUFFERED_IO

- Operating system: All
- Default=AUTOMATIC, Values: AUTOMATIC, ON, or OFF
- This variable allows you to control whether direct I/O (DIO) will be used on the log file system. When **DB2_LOGGER_NON_BUFFERED_IO** is set to AUTOMATIC, active log windows (namely, the primary log files) will be opened with DIO, and all other logger files will be buffered. When it is set to ON, all log file handles will be opened with DIO. When it is set to OFF, all log files handles will be buffered.

DB2MAXFSCRSEARCH

- Operating system: All
- Default=5, Values: -1, 1 to 33 554
- Specifies the number of free space control record (FSCRs) to search when adding a record to a table. The default is to search five FSCRs. Modifying this value allows you to balance insert speed with space reuse. Use large values to optimize for space reuse. Use small values to optimize for insert speed. Setting the value to -1 forces the database manager to search all FSCRs.

DB2_MAX_INACT_STMTS

- Operating system: All
- Default=Not set, Values: up to 4 000 000 000
- This variable overrides the default limit on the number of inactive statements kept by any one application. You can choose a different value in order to increase or reduce the amount of system monitor heap used for inactive statement information. The default limit is 250.

The system monitor heap can become exhausted if an application contains a very high number of statements in a unit of work, or if there are a large number of applications executing concurrently.

DB2_MAX_NON_TABLE_LOCKS

- Operating system: All
- Default=YES, Values: See description
- This variable defines the maximum number of NON table locks a transaction can have before it releases all of these locks. NON table locks are table locks that are kept in the hash table and transaction chain even when the transaction has finished using them. Because transactions often access the same table more than once, retaining locks and changing their state to NON can improve performance.

For best results, the recommended value for this variable is the maximum number of tables expected to be accessed by any connection. If no user-defined value is specified, the default value is as follows: If the **locklist** size is greater than or equal to

SQLP_THRESHOLD_VAL_OF_LRG_LOCKLIST_SZ_FOR_MAX_NON_LOCKS

(currently 8000), the default value is SQLP_DEFAULT_MAX_NON_TABLE_LOCKS_LARGE

(currently 150). Otherwise, the default value is SQLP_DEFAULT_MAX_NON_TABLE_LOCKS_SMALL

(currently 0).

DB2_MDC_ROLLOUT

- Operating system: All
- Default=IMMEDIATE, Values: IMMEDIATE, OFF, or DEFER
- This variable enables a performance enhancement known as "rollout" for deletions from MDC tables. Rollout is a faster way of deleting rows in an MDC table, when entire cells (intersections of dimension values) are deleted in a search DELETE statement. The benefits are reduced logging and more efficient processing.
- There are three possible outcomes of the variable setting:
 - No rollout if 0FF is specified
 - Immediate rollout if IMMEDIATE is specified.
 - Rollout with deferred index cleanup if DEFER is specified
- If the value is changed after startup, any new compilations of a statement will respect the new registry value setting. For statements that are in the package cache, no change in delete processing will be made until the statement is recompiled. The SET CURRENT MDC ROLLOUT MODE statement overrides the value of DB2_MDC_ROLLOUT at the application connection level.
- In Db2 Version 9.7 and later releases, the DEFER value is not supported for range-partitioned tables with partitioned RID indexes. Only the OFF and IMMEDIATE values are supported. The cleanup rollout type is IMMEDIATE if the DB2_MDC_ROLLOUT registry variable is set to DEFER, or if the CURRENT MDC ROLLOUT MODE special register is set to DEFERRED to override the DB2_MDC_ROLLOUT setting.

If only nonpartitioned RID indexes exist on the table, deferred index cleanup rollout is supported.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2MEMDISCLAIM

- Operating system: ALL
- Default=YES, Values: YES or NO
- Memory used by Db2 database system processes might have some associated paging space. This paging space might remain reserved even when the associated memory has been freed. Whether or not this is so depends on the operating system's (tunable) virtual memory management allocation policy. The **DB2MEMDISCLAIM** registry variable
controls whether Db2 agents explicitly request that the operating system disassociate the reserved paging space from the freed memory.

A **DB2MEMDISCLAIM** setting of YES results in smaller paging space requirements, and possibly less disk activity from paging. A **DB2MEMDISCLAIM** setting of N0 results in larger paging space requirements, and possibly more disk activity from paging. In some situations, such as if paging space is plentiful and real memory is so plentiful that paging never occurs, a setting of N0 provides a minor performance improvement.

DB2_MEM_TUNING_RANGE

- Operating system: All
- Default =NULL, Values: a sequence of percentages *n*, *m* where *n=minfree* and *m=maxfree* and *n < m*
- If this variable is not set, the Db2 database manager calculates values for *minfree* and *maxfree* based on the amount of memory on the server. In limited **instance_memory** environments, the Db2 database manager calculates values for *minfree* and *maxfree* based on the **instance_memory** setting. The setting of this variable has no effect unless the self-tuning memory manager (STMM) is enabled and **database_memory** is set to AUTOMATIC.

The *minfree* and *maxfree* settings represent the amount of instance memory, system memory or both that the STMM attempts to leave as a buffer. This buffer is critical to satisfying volatile memory requirements while avoiding memory over-commitment on the system or exhausting instance memory. In addition, the *minfree-maxfree* range is used to balance memory demands across multiple databases. In a single database that is tuned by STMM, the target free system or instance memory is always *minfree*. In a multiple database environment, the STMM tuner for the database with the highest memory demands targets the *minfree* value, while the STMM tuners for databases with lower demands have varying higher free memory targets (up to the *maxfree* value). The default *minfree, maxfree* settings are as follows:

Instance or System Memory		
Size	minfree (%)	maxfree(%)
1 GB	7.8	33
2 GB	7.4	29
4 GB	7.0	25
8 GB	6.7	22
16 GB	6.4	19
32 GB	6.2	17
64 GB	6.0	15
128 GB	5.8	13
256 GB	5.7	12
512 GB	5.6	11
1 TB	5.5	10

Table 5. Default minfree, maxfree settings

Starting with Version 10.5, a 5% extra buffer is added which is included in the values in the preceding table. This extra buffer is to accommodate the volatility of memory demands in a wider range of environments while maintaining the resilience expected of automatically tuned STMM environments. However, the additional 5% buffer is available to newly activating databases to minimize detuning (scaling back) upon activation. If a Version 10.5 or later STMM tuner detects the presence of STMM tuners from previous releases (that are competing with them for system memory), then the additional 5% buffer is removed from the calculation for the databases that are running on Version 10.5 or later. This removal of the additional 5% buffer, is to avoid biasing memory allocation towards the databases that are running on previous releases (which would tend to have lower free memory targets).

Performance gains might be achieved by reducing the *minfree, maxfree* settings in an STMM environment. However, care must be taken to ensure that volatility in memory requirements does not result in paging or memory exhaustion.

Changes to this variable take effect immediately for all STMM tuning operations. There is no need to restart the instance or to run the **db2set** command with the -immediate parameter.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2_MMAP_READ

- Operating system: AIX
- Default=0FF, Values: 0N or 0FF
- This variable is used in conjunction with **DB2_MMAP_WRITE** to allow the Db2 database system to use mmap as an alternate method of I/O.

When these variables are set to 0N, data is read into and written from the Db2 buffer pools using memory mapped I/O, and subsequently removed from the file system cache. This avoids double-caching of Db2 data. However, the recommended method to bypass the file system cache is to specify the NO FILE SYSTEM CACHING clause at the table space level, and to leave these variables at the default setting of 0FF.

DB2_MMAP_WRITE

- Operating system: AIX
- Default=0FF, Values: 0N or 0FF
- This variable is used in conjunction with **DB2_MMAP_READ** to allow the Db2 database system to use mmap as an alternate method of I/O.

When these variables are set to 0N, data is read into and written from the Db2 buffer pools using memory mapped I/O, and subsequently removed from the file system cache. This avoids double-caching of Db2 data. However, the recommended method to bypass the file system cache is to specify the NO FILE SYSTEM CACHING clause at the table space level, and to leave these variables at the default setting of 0FF.

DB2_NO_FORK_CHECK

- Operating system: UNIX
- Default=0FF, Values: 0N or 0FF
- When this variable is enabled, the Db2 runtime client minimizes checks to determine if the current process is a result of a fork call. This can improve performance of Db2 applications that do not use the fork() api.

DB2NTMEMSIZE

• Operating system: Windows

- Default= (varies by memory segment)
- Windows requires that all shared memory segments be reserved at DLL initialization time in order to guarantee matching addresses across processes. **DB2NTMEMSIZE** permits the user to override the Db2 defaults on Windows if necessary. In most situations, the default values should be sufficient. The memory segments, default sizes, and override options are:
 - 1. Parallel FCM Buffers: default size is 512 MB on 32-bit platforms, 4.5 GB on 64-bit platforms; override option is FCM:*number_of_bytes*
 - 2. Fenced Mode Communication: default size is 80 MB on 32-bit platforms, 512 MB on 64-bit platforms; override option is APLD:*number_of_bytes*
 - **3**. Message Query Memory: default size is 4 MB on 32-bit and 64-bit platforms; override option is QUE:<number of bytes>.

More than one segment may be overridden by separating the override options with a semicolon (;). For example, on a 32-bit version of Db2, to limit the FCM buffers to 1 GB, and the fenced stored procedures limit to 256 MB, use:

db2set DB2NTMEMSIZE=FCM:1073741824;APLD:268435456

To increase the message queue memory to 64 MB, use: db2set DB2NTMEMSIZE=QUE:67108864

DB2NTNOCACHE

- Operating system: Windows
- Default=0FF, Values: 0N or 0FF
- The **DB2NTNOCACHE** registry variable specifies whether the Db2 database system opens database files with a NOCACHE option. If **DB2NTNOCACHE** is set to 0N, file system caching is eliminated. If **DB2NTNOCACHE** is set to 0FF, the operating system caches Db2 files. This applies to all data except for files that contain long fields or LOBs. Eliminating system caching allows more memory to be available to the database so that the buffer pool or sort heap can be increased.

In Windows, files are cached when they are opened, which is the default behavior. One MB is reserved from a system pool for every 1 GB in the file. Use this registry variable to override the undocumented 192 MB limit for the cache. When the cache limit is reached, an out-of-resource error is given.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

Note: For table space containers, using the NO FILE SYSTEM CACHING clause with the ALTER TABLESPACE or CREATE TABLESPACE statement reports the same benefit as setting **DB2NTNOCACHE** to ON.

DB2NTPRICLASS

- Operating system: Windows
- Default=NULL, Values: R, H, (any other value)
- Sets the priority class for the Db2 instance (program DB2SYSCS.EXE). There are three priority classes:
 - NORMAL_PRIORITY_CLASS (the default priority class)
 - REALTIME_PRIORITY_CLASS (set by using R)

- HIGH_PRIORITY_CLASS (set by using H)

This variable is used in conjunction with individual thread priorities (set using **DB2PRIORITIES**) to determine the absolute priority of Db2 threads relative to other threads in the system.

Note: DB2NTPRICLASS is deprecated and should only be used at the recommendation of service. Use Db2 service classes to adjust agent priority and prefetch priority. Care should be taken when using this variable. Misuse could adversely affect overall system performance.

For more information, please refer to the SetPriorityClass() API in the Win32 documentation.

DB2NTWORKSET

- Operating system: Windows
- Default=1,1
- Used to modify the minimum and maximum working-set size available to the Db2 database manager. By default, when Windows is not in a paging situation, the working set of a process can grow as large as needed. However, when paging occurs, the maximum working set that a process can have is approximately 1 MB. **DB2NTWORKSET** allows you to override this default behavior.

Specify **DB2NTWORKSET** using the syntax **DB2NTWORKSET**=*min*, *max*, where *min* and *max* are expressed in megabytes.

DB2_OVERRIDE_BPF

- Operating system: All
- Default=Not set, Values: a positive numeric number of pages OR <entry>[;<entry>] where <entry> =<buffer pool ID>,<number of pages>
- This variable specifies the size of the buffer pool, in pages, to be created at database activation, rollforward recovery, or crash recovery. It is useful when memory constraints cause failures to occur during database activation, rollforward recovery, or crash recovery. The memory constraint could arise either in the rare case of a real memory shortage or, because of the attempt by the database manager to allocate a large buffer pool, in the case where there were inaccurately configured buffer pools. For example, when even a minimal buffer pool of 16 pages is not brought up by the database manager, try specifying a smaller number of pages using this environment variable. The value given to this variable overrides the current buffer pool size.

You can also use *<entry>*[;*<entry>*...] where *<entry>* =*<buffer pool ID>,<number of pages>* to temporarily change the size of all or a subset of the buffer pools so that they can start up.

DB2_PINNED_BP

- Operating system: AIX, HP-UX, Linux
- Default=N0, Values: YES or N0
- Setting this variable to YES causes Db2 to request that the Operating System pins Db2's Database Shared Memory. When configuring Db2 to pin Database Shared Memory, care should be taken to ensure that the system is not overcommitted, as the operating system will have reduced flexibility in managing memory.

On Linux, in addition to modifying this registry variable, the library, libcap.so.1 is also required.

Setting this variable to YES means that self tuning for database shared memory (activated by setting the **database_memory** configuration parameter to AUTOMATIC) cannot be enabled.

For HP-UX in a 64-bit environment, in addition to modifying this registry variable, the Db2 instance group must be given the MLOCK privilege. To do this, a user with root access rights performs the following actions:

- Adds the Db2 instance group to the /etc/privgroup file. For example, if the Db2 instance group belongs to db2iadm1 group then the following line must be added to the /etc/privgroup file: db2iadm1 MLOCK
- 2. Issues the following command:
 - setprivgrp -f /etc/privgroup

DB2PRIORITIES

- Operating system: All
- Values setting is platform dependent
- Controls the priorities of Db2 processes and threads.

Note: DB2PRIORITIES is deprecated and should only be used at the recommendation of service. Use Db2 service classes to adjust agent priority and prefetch priority.

DB2_RCT_FEATURES

- Operating system: All
- Default: NULL. Values: GROUPUPDATE=[ON|OFF]. The default value for GROUPUPDATE is OFF.
- This variable allows for optimized and reduced update processing for a searched UPDATE statement which targets multiple rows in an range clustered table when only equal predicates on the leading and subset of key sequence columns are specified. Logging is also reduced due to a single log record for all rows updated on a page, instead of a log record for each row updated.

Usage :

db2set DB2_RCT_FEATURES=GROUPUPDATE=ON

DB2_RESOURCE_POLICY

- Operating system: AIX, Linux, Windows
- Default=Not set, Values: valid path to configuration file (AIX, Linux, Windows) or AUTOMATIC (AIX, Linux, Windows)
- Defines a resource policy that can be used to control what operating system resources are used by the Db2 database, or contains rules for assigning specific operating system resources to specific Db2 database objects. For example, on AIX, Linux, or Windows operating systems, this registry variable can be used to limit the set of processors that the Db2 database system uses. The extent of resource control varies depending on the operating system.

On POWER7[®] systems running AIX 6.1 Technology Level (TL) 5 or higher, or any Linux or Windows operating systems, this variable can be set to AUTOMATIC. If you specify the AUTOMATIC option, the Db2 database system automatically determines the hardware topology and assigns engine dispatchable units (EDUs) to the various hardware modules in such a way that memory can be more efficiently shared between multiple EDUs that must access the same regions of memory. The AUTOMATIC setting also determines whether to enable memory affinitization, whereby EDUs attempt to allocate local memory during processing. This setting is intended for larger POWER7 systems with 16 or more cores or any Linux or Windows operating systems, and can enhance query performance for some workloads. To validate any performance improvements, it is best to run a performance analysis of the workload before and after you set the DB2_RESOURCE_POLICY variable to AUTOMATIC.

If you set the **DB2_RESOURCE_POLICY** parameter to AUTOMATIC, the database manager creates a resource group for every set of related resources that is visible through the AIX topology interfaces. The Db2 instance must be running on a POWER7 machine with enhanced affinity support, which is available and enabled by default on AIX 6.1 TL5 or higher. On other supported platforms, the database manager creates one resource group and makes all resources part of that group.

You can set the registry variable to indicate the path to a configuration file that defines a policy for binding Db2 processes to operating system resources. The resource policy enables you to specify a set of operating system resources to restrict the Db2 database system. Each Db2 process is bound to a single resource of the set. Resource assignment occurs in a circular round robin fashion. Under certain types of workloads, a resource policy can be beneficial on hardware that exhibits NUMA properties.

Sample configuration files:

Example 1: Bind all Db2 processes to either CPU 1 or 3.

<RESOURCE_POLICY>

<GLOBAL_RESOURCE_POLICY>
<METHOD>CPU</METHOD>
<RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE_BINDING>
<RESOURCE_S3</RESOURCE>
</RESOURCE_BINDING>
</GLOBAL_RESOURCE_POLICY>
</RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></RESOURCE_POLICY></POLICY></POLICY></POLICY></POLICY></POLICY></POLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY></poNUPLICY>

Example 2: (AIX only) Bind Db2 processes to one of the following resource sets: sys/node.03.00000, sys/node.03.00001, sys/node.03.00002, sys/node.03.00003

```
<RESOURCE POLICY>
  <GLOBAL RESOURCE POLICY>
   <METHOD>RSET</METHOD>
   <RESOURCE BINDING>
   <RESOURCE>sys/node.03.00000</RESOURCE>
   </RESOURCE BINDING>
   <RESOURCE BINDING>
   <RESOURCE>sys/node.03.00001</RESOURCE>
   </RESOURCE BINDING>
   <RESOURCE BINDING>
   <RESOURCE>sys/node.03.00002</RESOURCE>
   </RESOURCE BINDING>
   <RESOURCE BINDING>
   <RESOURCE>sys/node.03.00003</RESOURCE>
   </RESOURCE_BINDING>
  </GLOBAL_RESOURCE_POLICY>
 </RESOURCE POLICY>
```

Note: For AIX only, use of the RSET method requires CAP_NUMA_ATTACH and CAP_PROPAGATE capabilites.

Example 3: (Linux only) Bind all memory from bufferpool IDs 2 and 3 which are associated with the SAMPLE database to NUMA node 3. Also use 80 percent of the total database memory for the binding to NUMA node 3 and leave 20 percent to be striped across all nodes for non-bufferpool specific memory.

<RESOURCE POLICY>

<DATABASE_RESOURCE_POLICY>
<DBNAME>samp1e</DBNAME>
<METHOD>NODEMASK</METHOD>
<RESOURCE_BINDING>
<RESOURCE>3</RESOURCE>
<DBMEM_PERCENTAGE>80</DBMEM_PERCENTAGE>
<BUFFERPOOL_BINDING>
<BUFFERPOOL_ID>2</BUFFERPOOL_ID>
<BUFFERPOOL_BINDING>
</RESOURCE_BINDING>
</RESOURCE_BINDING>
</RESOURCE_BINDING>
</RESOURCE_POLICY>

Example 4: (For Linux and Windows only) Define two distinct processor sets specified by CPU masks 0x0F and 0xF0. Bind Db2 processes and bufferpool ID 2 to processor set 0x0F and Db2 processes and bufferpool ID 3 to processor set 0xF0. For each processor set, use 50 percent of the total database memory for the binding.

This resource policy is useful when a mapping between processors and NUMA nodes is desired. An example of such a scenario is a system with 8 processors and 2 NUMA nodes where processors 0 to 3 belong to NUMA node 0 and processors 4 to 7 belong to NUMA node 1. This resource policy allows for processor binding while implicitly maintaining memory locality (that is, a hybrid of CPU method and NODEMASK method).

```
<RESOURECE POLICY>
  <DATABASE RESOURCE POLICY>
    <DBNAME>sample</DBNAME>
    <METHOD>CPUMASK</METHOD>
    <RESOURCE BINDING>
      <RESOURCE>0x0F</RESOURCE>
      <DBMEM PERCENTAGE>50</DBMEM PERCENTAGE>
      <BUFFERPOOL BINDING>
        <BUFFERPOOL ID>2</BUFFERPOOL ID>
      </BUFFERPOOL BINDING>
    </RESOURCE BINDING>
    <RESOURCE BINDING>
      <RESOURCE>0x0F</RESOURCE>
      <DBMEM_PERCENTAGE>50</DBMEM_PERCENTAGE>
      <BUFFERPOOL BINDING>
        <BUFFERPOOL ID>3</BUFFERPOOL ID>
      </BUFFERPOOL BINDING>
    </RESOURCE BINDING>
  </DATABASE RESOURCE POLICY>
</RESOURCE POLICY>
```

Example 5: (AIX operating systems only) You can manually enable resource group awareness by specifying resource groups in the resource policy file. Use the RESOURCE_GROUP element to specify the resources that belong to a particular resource group. The defined resource groups do not have to align with NUMA boundaries. The RESOURCE_GROUP column of the ENV_GET_DB2_EDU_SYSTEM_RESOURCES table function identifies the resource group with which an EDU is associated. Define two resource groups, each of which contains four scheduler resource affinity domains (SRADs): <RESOURCE POLICY> <GLOBAL RESOURCE POLICY> <METHOD>SRAD</METHOD> <RESOURCE GROUP> <RESOURCE_GROUP_NAME>TESTGROUP1</RESOURCE_GROUP_NAME> <RESOURCE BINDING> <RESOURCE>0</RESOURCE> </RESOURCE_BINDING> <RESOURCE BINDING> <RESOURCE>1</RESOURCE> </RESOURCE BINDING> <RESOURCE BINDING> <RESOURCE>2</RESOURCE> </RESOURCE BINDING> <RESOURCE BINDING> <RESOURCE>3</RESOURCE> </RESOURCE BINDING> </RESOURCE GROUP> <RESOURCE GROUP> <RESOURCE GROUP NAME>TESTGROUP2</RESOURCE GROUP NAME> <RESOURCE BINDING> <RESOURCE>4</RESOURCE> </RESOURCE BINDING> <RESOURCE BINDING> <RESOURCE>5</RESOURCE> </RESOURCE_BINDING> <RESOURCE BINDING> <RESOURCE>6</RESOURCE> </RESOURCE BINDING> <RESOURCE BINDING> <RESOURCE>7</RESOURCE> </RESOURCE BINDING> </RESOURCE_GROUP> </GLOBAL RESOURCE POLICY> </RESOURCE POLICY> The configuration file specified by the DB2_RESOURCE_POLICY registry variable accepts a SCHEDULING POLICY element. You can use the SCHEDULING_POLICY element on some platforms to select - The operating system scheduling policy used by the Db2 server You can set an operating system scheduling policy for Db2 on AIX, and for Db2 on Windows using the **DB2NTPRICLASS** registry variable. - The operating system priorities used by individual Db2 server agents Alternatively, you can use the registry variables **DB2PRIORITIES** and **DB2NTPRICLASS** to control the operating system scheduling policy and set Db2 agent priorities. However, the specification of a

SCHEDULING_POLICY element in the resource policy configuration file provides a single place to specify both the scheduling policy and the associated agent priorities.

Example 1: Selection of the AIX SCHED_FIFO2 scheduling policy with a priority boost for the Db2 log writer and reader processes.

```
<RESOURCE_POLICY>

<SCHEDULING_POLICY>

<POLICY_TYPE>SCHED_FIF02</POLICY_TYPE>

<PRIORITY_VALUE>60</PRIORITY_VALUE>

<EDU_PRIORITY>

<EDU_NAME>db2loggr</EDU_NAME>

<PRIORITY_VALUE>56</PRIORITY_VALUE>

</EDU_PRIORITY>

<EDU_PRIORITY>

<EDU_PRIORITY>

<EDU_PRIORITY>

<SCHEDULING_POLICY>

</RESOURCE_POLICY>
```

Example 2: Replacement for DB2NTPRICLASS=H on Windows.

```
<RESOURCE_POLICY>

<SCHEDULING_POLICY>

<POLICY_TYPE>HIGH_PRIORITY_CLASS</POLICY_TYPE>

</SCHEDULING_POLICY>

</RESOURCE_POLICY>
```

DB2_SELUDI_COMM_BUFFER

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- This variable is used during the processing of blocking cursors over SELECT from UPDATE, INSERT, or DELETE (UDI) queries. When enabled, this registry variable prevents the result of a query from being stored in a temporary table. Instead, during the OPEN processing of a blocking cursor for a SELECT from UDI query, the Db2 database system attempts to buffer the entire result of the query directly into the communications buffer memory area.

Note: If the communications buffer space is not large enough to hold the entire result of query, an SQLCODE -906 error is issued, and the transaction is rolled back. See the **aslheapsz** and **rqrioblk** database manager configuration parameters for information on adjusting the size of the communication buffer memory area for local and remote applications respectively.

This registry variable is not supported when intrapartition parallelism is enabled.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_SET_MAX_CONTAINER_SIZE

- Operating system: All
- Default=Not set, Values: -1, any positive integer greater than 65 536 bytes

• This registry variable allows you to limit the size of individual containers for automatic storage table spaces with the AutoResize feature enabled.

Note: Although you can specify **DB2_SET_MAX_CONTAINER_SIZE** in bytes, kilobytes, or megabytes, **db2set** indicates its value in bytes.

• If the value is set to -1, there will be no limit to the size of a container.

DB2_SKIPDELETED

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- When enabled, this variable allows statements using either Cursor Stability or Read Stability isolation levels to unconditionally skip deleted keys during index access and deleted rows during table access. With DB2_EVALUNCOMMITTED enabled, deleted rows are automatically skipped, but uncommitted pseudo-deleted keys in indexes are not skipped unless DB2_SKIPDELETED is also enabled. Only scans that are not Currently Committed (CC) applicable considers these variables.

DB2_SKIPDELETED is only applicable when currently committed semantics will not help avoid lock contentions. When this variable is set and currently committed is applicable to a scan, deleted rows will not be skipped; their currently committed version will be processed instead

This registry variable does not impact the behavior of cursors on the Db2 catalog tables.

This registry variable is activated with the **db2start** command.

DB2_SKIPINSERTED

- Operating system: All
- Default=0FF, Values: 0N or 0FF
- When the **DB2_SKIPINSERTED** registry variable is enabled, it allows statements using either Cursor Stability or Read Stability isolation levels to skip uncommitted inserted rows as if they had not been inserted. This registry variable does not impact the behavior of cursors on the Db2 catalog tables. This registry variable is activated at database startup, while the decision to skip uncommitted inserted rows is made at statement compile or bind time.

This registry variable has no effect if currently committed semantics are being used. That is, even if **DB2_SKIPINSERTED** is set to 0FF and currently committed behavior is enabled, uncommitted inserted rows are still skipped.

Note: Skip inserted behavior is not compatible with tables that have pending rollout cleanup. As a result, scanners might wait for locks on a RID only to discover that the RID is part of a rolled out block.

DB2_SMP_INDEX_CREATE

- Operating system: All
- Default=Not set, Values: 2 to 1000
- This dynamic registry variable overrides the default number of agents used to scan and sort the index data when building or rebuilding an index. This registry variable is only checked when the index manager component determines that parallelism is warranted. That decision is based on many considerations, including table size and whether multiple processors are present.

DB2_SMP_INDEX_CREATE has an effect only when it is set to a non-zero value. When increasing the number of agents used to scan and sort the index data, it is important to ensure that the database configuration parameters **sortheap** and **sheapthres_shr** are set appropriately. The more memory available for sorting (specified by the **sheapthres_shr** parameter), the less likely sorting the index data requires writing out temporary results to a system temporary tablespace. If the sort does not spill to disk, it is much faster. Furthermore, to ensure that each agent participating in the sort gets an equal amount of memory, the **sortheap** parameter should be set to a value no greater than **sheapthres_shr**/*n*, where *n* is the number of agents used to scan and sort the index table.

DB2_SMS_TRUNC_TMPTABLE_THRESH

v98_u3

- Operating system: All
- Default=-2, Values: -2, -1, 0to *n*, where *n*=the number of extents per temporary table in the SMS table space container that are to be maintained
- This variable specifies a minimum file size threshold at which the file representing a temporary table is maintained in SMS table spaces.

The default setting for this variable is -2, which means that there will not be any unnecessary file system access for any spilled SMS temporary objects whose size is less than or equal to 1 extent * number of containers. Temporary objects that are larger than this are truncated to 0 extent.

When this variable is set to 0, no special threshold handling is done. Instead, once a temporary table is no longer needed, that file is truncated to 0 extent. When the value of this variable is greater than 0, a larger file is maintained. Objects larger than the threshold will be truncated to the threshold size. This reduces some of the system overhead involved in dropping and recreating the file each time a temporary table is used.

If this variable is set to -1, the file is not truncated and the file is allowed to grow indefinitely, restricted only by system resources.

DB2_SORT_AFTER_TQ

- Operating system: All
- Default=N0, Values: YES or N0
- Specifies how the optimizer works with directed table queues in a partitioned database environment when the receiving end requires the data to be sorted and the number of receiving nodes is equal to the number of sending nodes.

When **DB2_SORT_AFTER_TQ=**NO, the optimizer tends to sort at the sending end and merge the rows at the receiving end.

When **DB2_SORT_AFTER_TQ**=YES, the optimizer tends to transmit the rows unsorted, not merge at the receiving end, and sort the rows at the receiving end after receiving all the rows.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_SQLWORKSPACE_CACHE

• Operating system: All

- Default: 30, Values: 10 2000
- This variable allows you to control the amount of caching of previously used sections in the SQL Workspace.

The SQL Workspace contains allocations, in the form of sections, for the execution of SQL. Each SQL statement (static or dynamic) that is being executed on behalf of an application must maintain a unique copy of the section in the SQL Workspace for the duration of execution of that statement. Once the execution of the statement is complete, the section becomes inactive and the memory allocations associated with an inactive section can either be freed, or they can remain cached in the SQL Workspace. When a new execution of the same SQL statement occurs from any connection, it may find a cached copy of the section in the SQL Workspace left from a previous execution, thus saving the costs associated with allocating and initializing a new copy of the section. In such a manner, the SQL Workspace contains both active sections, corresponding to currently executing SQL, and cached sections that are not currently executing.

The value for this registry variable specifies the percentage of memory allocations that are allowed to remain cached in the SQL Workspace. This caching is expressed as a percentage of the memory allocations for active sections. Thus, for example, a value of 50 would mean that the SQL workspace contains all of the active (currently executing) sections and up to 50% more of previously executed cached sections that can be reused. You would adjust the setting for DB2_SQLWORKSPACE_CACHE based on how much of the SQL workspace you want to make available for reuse. For example, increasing the size of this variable, can result in some performance improvements for OLTP workloads. On the other hand, a higher setting also means that there is an increase in the size of the application shared heap.

Note: if the appl_memory database configuration parameter is not set to AUTOMATIC, the size of the SQL Workspace may also be limited by the appl_memory and the SQL Workspace may not provide as much caching as the DB2_SQLWORKSPACE_CACHE setting might allow for; you might want to consider increasing appl_memory (or setting it to AUTOMATIC) in such a case.

This registry variable is not dynamic

DB2_TRUST_MDC_BLOCK_FULL_HINT

- Operating system: All
- Default: OFF, Values: 0N or 0FF
- When you insert records into an MDC table, Db2 searches the composite block index for blocks that have the same dimension values as the new record that is being inserted. Those blocks are then checked to determine whether they have enough free space for the new record. For any block checked that does not have enough free space, Db2 sets the Full_Block bit in the composite block index for that block. If the list of blocks for a specified dimension value is long and most of those blocks are full, then a significant amount of time is spent searching.

When the **DB2_TRUST_MDC_BLOCK_FULL_HINT** variable is set, Db2 skips searching for free space in any block that is marked with the **Full_Block** bit in the composite block index. This **Full_Block** bit is only a hint as the bit is only cleared when the entire block is removed and when the composite block index is rebuilt by using the **REORG** command. The trade-off is that some free space might be wasted if deletes are run that partially empties blocks as oppose to fully emptying them with rollout delete. For more information about rollout deletes, see "Rollout deletion" in the "Optimization strategies for MDC tables" topic.

DB2_TRUSTED_BINDIN

- Operating system: All
- Default=0FF, Values: 0FF, 0N, or CHECK
- When **DB2_TRUSTED_BINDIN** is enabled, it speeds up the execution of query statements containing host variables within an embedded unfenced stored procedure.

When this variable is enabled, there is no conversion from the external SQLDA format to an internal Db2 format during the binding of SQL and XQuery statements contained within an embedded unfenced stored procedure. This will speed up the processing of the embedded SQL and XQuery statements.

The following data types are not supported in embedded unfenced stored procedures when this variable is enabled:

- SQL_TYP_DATE
- SQL_TYP_TIME
- SQL_TYP_STAMP
- SQL_TYP_CGSTR
- SQL_TYP_BLOB
- SQL_TYP_CLOB
- SQL_TYP_DBCLOB
- SQL_TYP_CSTR
- SQL_TYP_LSTR
- SQL_TYP_BLOB_LOCATOR
- SQL_TYP_CLOB_LOCATOR
- SQL_TYP_DCLOB_LOCATOR
- SQL_TYP_BLOB_FILE
- SQL_TYP_CLOB_FILE
- SQL_TYP_DCLOB_FILE
- SQL_TYP_BLOB_FILE_OBSOLETE
- SQL_TYP_CLOB_FILE_OBSOLETE
- SQL_TYP_DCLOB_FILE_OBSOLETE

If these data types are encountered, an SQLCODE -804, SQLSTATE 07002 error is returned.

Note: The data type and length of the input host variable must match the internal data type and length of the corresponding element exactly. For host variables, this requirement will always be met. However, for parameter markers, care must be taken to ensure that matching data types are used. The CHECK option can be used to ensure that the data types and lengths match for all input host variables, but this option negates most of the performance improvements.

Note: DB2_TRUSTED_BINDIN is deprecated and will be removed in a later release.

DB2_USE_ALTERNATE_PAGE_CLEANING

• Operating system: All

- Default=Not set, Values: ON or OFF
- This variable specifies whether a Db2 database uses the alternate method of page cleaning algorithms or the default method of page cleaning. When this variable is set to 0N, the Db2 system writes changed pages to disk, keeping ahead of LSN_GAP and proactively finding victims. Doing this allows the page cleaners to better utilize available disk I/O bandwidth. When this variable is set to 0N, the *chngpgs_thresh* database configuration parameter is no longer relevant because it does not control page cleaner activity.

DB2_USE_FAST_PREALLOCATION

- Operating system: AIX and Linux on Veritas VxFS, JFS2, GPFS[™], ext4 (Linux only) and xFS (Linux only) file systems
- Default: ON for Veritas VxFS, JFS2, GPFS, ext4 and xFS, Values: ON or OFF
- Allows the fast preallocation file system feature to reserve table space, and speed up the process of creating or altering large table spaces and database restore operations. This speed improvement is implemented at a small delta cost of performing actual space allocation during runtime when rows are inserted.

To disable fast preallocation, set **DB2_USE_FAST_PREALLOCATION** to OFF. This might improve runtime performance, at the cost of slower table space creation and database restore times, on some operating systems, especially AIX, when there is a large volume of inserts and selects on same table space. Note that once fast preallocation is disabled, the database has to be restarted.

DB2_USE_FAST_LOG_PREALLOCATION

- Operating system: AIX and Linux on Veritas VxFS, JFS2, GPFS, ext4 (Linux only) and xFS (Linux only) file systems
- Default: OFF, ON under DB2_WORKLOAD=SAP, Values: ON or OFF
- Allows the fast preallocation file system feature to reserve space for log files, and speed up the process of creating or altering large log files, if the underlying file system supports this feature. This speed improvement is implemented at a small delta cost of performing actual space allocation during runtime when log records get written to such preallocated log files.

To enable fast preallocation for logs, set **DB2_USE_FAST_LOG_PREALLOCATION** to ON.

DB2_USE_IOCP

- Operating system: AIX
- Default=ON, Values: ON or OFF
- This variables enables the use of AIX I/O completion ports (IOCP) when submitting and collecting asynchronous I/O (AIO) requests. This feature is used to enhance performance in a non-uniform memory access (NUMA) environment by avoiding remote memory access.

Miscellaneous variables

You can use Db2 variables to control and maintain your database. When you set variables you can control areas such as the default administration server, the default client path, and the ability to commit changes that are made to data when you exit an application.

DB2ADMINSERVER

- Operating system: Windows and UNIX
- Default: NULL
- Specifies the Db2 Administration Server.

DB2_ATS_ENABLE

- Operating system: All
- Default: NULL, Values: YES/TRUE/ON/1 or NO/FALSE/OFF/0
- This variable controls whether the administrative task scheduler is running. The administrative task scheduler is disabled by default. When the scheduler is disabled, you can use the built-in procedures and views to define and modify tasks but the scheduler will not execute the tasks.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2AUTH

- Operating system: All
- Default: Not set. Values: DISABLE_CHGPASS, OSAUTHDB, SQLADM_NO_RUNSTATS_REORG, TRUSTEDCLIENT_DATAENC, TRUSTEDCLIENT_SRVRENC, DISABLE_SYSMON_CONNECT, JCC_ENFORCE_SECMEC, JCC_NOENFORCE_SECMEC_MSG, JCC_NOENFORCE_SECMEC_NOMSG.
- This variable allows you to tune the behavior of user authentication. Valid values are as follows:
 - ALLOW_LOCAL_FALLBACK: This value allows the Db2 server to fall back to using SERVER authentication for local implicit connects or attaches when the server is configured to use Generic Security Service (GSS) plugins. When ALLOW_LOCAL_FALLBACK is enabled, for local implicit connects, the userid and password plugin specified by the srvcon_pw_plugindatabase manager configuration parameter is used for authenticating the user, instead of using the specified GSS authentication, such as KERBEROS, KRB_SERVER_ENCRYPT, GSSPLUGIN, or GSS_SERVER_ENCRYPT.

A local implicit connect is created when you issue a connect to a local database (note that *local* means only IPC, not TCP/IP) without providing a userid or password. Db2 uses the userid of the current session or process for the userid of the connect. The password plugins provided by Db2 assume that a userid retrieved from the operating system has already been authenticated by the operating system and, therefore, password validation is not necessary.

Note: If you provide a userid and password, it is not considered a local implicit connect and the ALLOW_LOCAL_FALLBACK option does not apply.

The password plugin Db2 uses is determined by the **srvcon_pw_plugin** database manager configuration parameter. If the **srvcon_pw_plugin** parameter is set to IBMLDAPauthserver, the IBMLDAPauthserver plugin processes the local implicit connect. If the **srvcon_pw_plugin** parameter is set to a custom security plugin, the custom plugin processes the local implicit connect. If the **srvcon_pw_plugin** parameter is not set, the default plugin (IBMOSauthserver) processes the local implicit connect. The security plugins provided by your Db2 database product always allow a local implicit connect because they assume that the user has been validated by the OS.

- DISABLE_CHGPASS: This value disables the ability to change the password from the client.
- OSAUTHDB: This value instructs the Db2 database manager to use the authentication and group setting for a user on the AIX operating system. Transparent LDAP support has also been extended to the Linux, HP-UX and Solaris operating systems. The LDAP server can be any one of the following:
 - IBM Tivoli Directory Server (ITDS)
 - Microsoft Active Directory (MSAD)
 - Sun One Directory Server
- SQLADM_NO_RUNSTATS_REORG: This value, introduced in Db2 Version 9.7 Fix Pack 5, disables the ability of users with SQLADM authority to perform a reorg or runstats operation.
- TRUSTEDCLIENT_DATAENC: This value forces untrusted clients to use DATA_ENCRYPT. This value is not applicable to a Db2 Connect gateway.
- TRUSTEDCLIENT_SRVRENC: This value forces untrusted clients to use SERVER_ENCRYPT. This value is not applicable to a Db2 Connect gateway.
- JCC_ENFORCE_SECMEC: This value, introduced in Db2 10.5.0.4, forces the Db2 server to not accept CLEAR_TEXT_PASSWORD_SECURITY security mechanisms when the authentication value is set to SERVER_ENCRYPT. Cannot be set together with either JCC_NOENFORCE_SECMEC_MSG or JCC_NOENFORCE_SECMEC_NOMSG settings.
- JCC_NOENFORCE_SECMEC_MSG: This value, introduced in Db2 11.1.3.3, allows the Db2 server to accept
 CLEAR_TEXT_PASSWORD_SECURITY security mechanisms when the authentication value is set to SERVER_ENCRYPT, and prints a warning in the db2diag.log regarding the associated risk of sending the password in the clear. If set, the following message is printed on every connect: Connection accepted as SERVER (JCC_NOENFORCE_SECMEC_MSG). Cannot be set together with either JCC_ENFORCE_SECMEC or JCC_NOENFORCE_SECMEC_NOMSG settings.
- JCC_NOENFORCE_SECMEC_NOMSG: This value, introduced in Db2 11.1.3.3, allows the Db2 server to accept
 CLEAR_TEXT_PASSWORD_SECURITY security mechanisms when the authentication value is set to SERVER_ENCRYPT, but does not print a warning in the db2diag.log regarding the associated risk of sending the password in the clear. Cannot be set together with either JCC_ENFORCE_SECMEC or JCC_NOENFORCE_SECMEC_NOMSG settings. This value represents the default behavior of Db2 since Db2 10.5.0.4.
- DISABLE_SYSMON_CONNECT: This value removes the implicit privilege to connect to the database from SYSMON.
- You can set multiple values for **DB2AUTH** using the **db2set** command. You must separate each value with a comma. For example, if you want **DB2AUTH** to have both DISABLE_CHGPASS and OSAUTHDB enabled, use the command:

db2set DB2AUTH=DISABLE_CHGPASS,OSAUTHDB

Note: You cannot enable TRUSTEDCLIENT_SRVRENC and TRUSTEDCLIENT_DATAENC at the same time.

DB2_BCKP_COMPRESSION

• Operating system: All

- Default: COMPRESS, Values: COMPRESS, NX842
- Specifies whether backups are compressed with the common Db2 compression or with NX842 compression. NX842 compression is only available on AIX (see Hardware accelerated backup and log file compression for details).

DB2_BCKP_INCLUDE_LOGS_WARNING

- Operating system: All
- Default: FALSE, Values: FALSE, TRUE
- Specifies whether online backups which fail to include all of the necessary log files should still be allowed to complete successfully. By default, online backups that do not explicitly specify either the INCLUDE LOGS or the EXCLUDE LOGS option fail if all of the logs are not successfully included. When this variable is set to TRUE, these backups will be allowed to succeed with a warning.

In SAP environments, when **DB2_WORKLOAD**=SAP is set, the default value of this registry variable is TRUE.

DB2_BCKP_PAGE_VERIFICATION

- Operating system: All
- Default: FALSE, Values: FALSE, TRUE
- Specifies whether DMS and AS page validation occurs during a backup. (See the Usage Notes of the Backup Command for more details.)

DB2CLIINIPATH

- Operating system: All
- Default: NULL
- Used to override the default path of the CLI/ODBC configuration file (db2cli.ini) and specify a different location on the client. The value specified must be a valid path on the client system.

DB2_COMMIT_ON_EXIT

- Operating system: UNIX
- Default: 0FF, Values: 0FF/N0/0 or 0N/YES/1
- On UNIX operating systems, prior to Db2 UDB Version 8, Db2 committed any remaining in-flight transactions on successful application exit.
- In Db2 UDB Version 8, the behavior was changed so that in-flight transactions were rolled back on exit. This registry variable allows users with embedded SQL applications which depend on the earlier behavior to continue to enable it in Db2 Version 9. This registry variable does not affect JDBC, CLI, and ODBC applications.

Note that this registry variable is deprecated, and the commit-on-exit behavior will no longer be supported in future release. Users should determine whether any of their applications developed prior to Db2 Version 9 continue to depend on this functionality, and add the appropriate explicit COMMIT or ROLLBACK statements to the application as required. If the registry variable is turned on, care should be taken not to implement new applications which fail to explicitly COMMIT before exit.

Most users should leave this registry variable at the default setting.

DB2_COMMON_APP_DATA_PATH

• Operating system: Windows

- Default: Windows' C:\ProgramData\
- Points to user-defined location that holds the Db2 common application data for the Db2 copy. This registry variable is populated if **DB2_COMMON_APP_DATA_TOP_PATH** is specified during the response file installation or if "Db2 Common Application Data Top Path" field is populated during the custom installation step.

Starting in Version 9.7 Fix Pack 5, this registry variable is visible in **db2set** command output but is not changeable. Any attempts to change given registry value will result in an error.

DB2_COMPATIBILITY_VECTOR

- Operating system: All
- Default: NULL, Values: NULL or 00 to FFF
- The DB2_COMPATIBILITY_VECTOR registry variable is used to enable one or more Db2 compatibility features introduced since Db2 Version 9.5.
- These features ease the task of migrating applications written for other relational database vendors to Db2 Version 9.5 or later.
- DB2_COMPATIBILITY_VECTOR is represented as a hexadecimal value, and each bit in the variable enables one of the Db2 compatibility features as outlined in the DB2_COMPATIBILITY_VECTOR values table. To enable all of the supported compatibility features, set the registry variable to the value ORA (which is equivalent to the hexadecimal value FFF). This is the recommended setting.

DB2CONNECT_DISCONNECT_ON_INTERRUPT

- Operating system: All
- Default: NO, Values: YES/TRUE/1 or NO/FALSE/0
- When set to YES (TRUE or 1), this variable specifies that the connection to a Version 8 (or higher) Db2 Universal Database z/OS server should be broken immediately when an interrupt occurs. You can use this variable in the following configurations:
 - If you are running a Db2 client with a Version 8 (or higher) Db2 z/OS server, set DB2CONNECT_DISCONNECT_ON_INTERRUPT to YES on the client.
 - If you are running a Db2 client through a Db2 Connect gateway to a Version 8 (or higher) Db2 z/OS server, set
 DB2CONNECT_DISCONNECT_ON_INTERRUPT to YES on the gateway.

DB2_CREATE_DB_ON_PATH

- Operating system: Windows
- Default: NULL, Values: YES or NO
- Set this registry variable to YES to enable support for the use of a path (as well as a drive) as a database path. The setting of **DB2_CREATE_DB_ON_PATH** is checked when a database is created, when the database manager configuration parameter **dftdbpath** is set, and when a database is restored. The fully qualified database path can be up to 215 characters in length.

If **DB2_CREATE_DB_ON_PATH** is not set (or is set to N0) and you specify a path for the database path when creating or restoring a database, error SQL1052N is returned.

If **DB2_CREATE_DB_ON_PATH** is not set (or is set to N0) and you update the **dftdbpath** database manager configuration parameter, error SQL5136N is returned.

CAUTION:

If path support is used to create new databases, applications written prior to Db2 Version 9.1 using the db2DbDirGetNextEntry() API or an older version of it, might not work correctly. Please refer to http://www.ibm.com/software/data/db2/support/db2_9/ for details on various scenarios and the proper course of action.

DB2_CREATE_INDEX_ALLOW_WRITE

- Operating system: All
- Default: ON (OFF for pureScale), Values: ON, OFF
- This registry variable overrides the default behavior with respect to whether applications are allowed to write access to the table. The default behavior is ON (which means other applications have write access to the table) for non-pureScale, and OFF for pureScale.

Note: There is a limitation on creating expression-based indexes in pureScale, during which application can only have read access, even if this registry variable is set.

- This variable is available in Version 11.1.3.3 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

DB2_DDL_SOFT_INVAL

- Operating system: All
- Default: ON, Values: ON or OFF
- Enables soft invalidation of applicable database objects when they are dropped or altered.

When **DB2_DDL_SOFT_INVAL** is set to 0N, any DDL operation, such as drop, alter, or detach, can start without waiting for transactions referencing the same objects to finish. Current executions dependant on the objects will continue with the original object definition, while new executions will utilize the changed object. This allows for better concurrency when issuing DDL statements.

Note: The new soft invalidation capabilities only apply to dynamic packages. Any objects with static packages will still require a hard invalidation.

DB2_DISABLE_FLUSH_LOG

- Operating system: All
- Default: OFF, Values: ON or OFF
- Specifies whether to disable closing the active log file when the online backup is completed.

When an online backup completes, the last active log file is truncated, closed, and made available to be archived. This ensures that your online backup has a complete set of archived logs available for recovery. You might want to disable closing the last active log file if you are concerned that you are wasting log space.

You might also want to disable closing the last active log file if you find you are receiving log full messages a short time after the completion of the online backup. When a log file is truncated, the reserved active log space is incremented by the amount proportional to the size of the truncated log. The active log space is freed once the truncated log file is reclaimed. The reclamation occurs a short time after the log file becomes inactive. During the short interval between these two events, you may receive log full messages.

During any backup which includes logs, this registry variable is ignored, since the active log file must be truncated and closed in order for the backup to include the logs.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2_DISPATCHER_PEEKTIMEOUT

- Operating system: All
- Default: 1, Values: 0 to 32767 seconds; 0 denotes that timeout is immediate
- DB2_DISPATCHER_PEEKTIMEOUT allows you to adjust the time, in seconds, that a dispatcher waits for a client's connection request before handing the client off to an agent. In most cases, you should not need to adjust this registry variable. This registry variable only affects instances that have Db2 Connect connection concentrator enabled.

This registry variable and the DB2_SERVER_CONTIMEOUT registry variable both configure the handling of a new client during connect time. If there are many slow clients connecting to an instance, the dispatcher may be held up for up to 1 second to timeout each client, causing the dispatcher to become a bottle neck, if many clients are connecting simultaneously. If an instance with multiple active databases is experiencing very slow connection times, DB2_DISPATCHER_PEEKTIMEOUT may be lowered to 0. Lowering DB2_DISPATCHER_PEEKTIMEOUT causes the dispatcher to only look into the client's connect request if it is already there; the dispatcher will not wait for the connect request to arrive. If an invalid value is set, the default value is used. This registry variable is not dynamic.

DB2_DJ_INI

- Operating system: All
- Default:
- When its value is as default, Federation will pick up db2dj.ini file from below locations:
 - UNIX: db2_instance_directory/cfg/db2dj.ini
 - Windows: db2_install_directory\cfg\db2dj.ini
- Specifies the absolute path name of the federation configuration file, for example: db2set DB2_DJ_INI=\$HOME/sqllib/cfg/my_db2dj.ini This file contains the settings for data source environment variables. These environment variables are used by the Informix[®] wrapper and by the wrappers provided by InfoSphere[®] Federation Server.

Here is a sample federation configuration file:

```
INFORMIXDIR=/informix/client_sdk
INFORMIXSERVER=inf93
ORACLE_HOME=/usr/oracle9i
SYBASE=/sybase/V12
SYBASE_OCS=0CS-12_5
```

The following restrictions apply to the db2dj.ini file:

- Entries must follow the format *evname=value* where *evname* is the name of the environment variable and *value* is its value.
- The environment variable name has a maximum length of 255 bytes.

- The environment variable value has a maximum length of 765 bytes. This variable is ignored unless the database manager parameter **federated** is set to YES.

DB2_DMU_DEFAULT

- Operating system: All
- Default: NULL, Values: : IMPLICITLYHIDDENMISSING, IMPLICITLYHIDDENINCLUDE
- This variable allows you to set the default behavior of whether implicitly hidden columns are included when the column list is omitted by the load, import, ingest, and export utilities. Valid values are as follows:

NULL

It means that no default behavior is specified. If the table has implicitly hidden columns, the column list must be explicitly specified or the hidden column options must be specified by the utilities. Otherwise an error occurs.

IMPLICITLYHIDDENMISSING

The utilities assume that the implicitly hidden columns are not included by default unless the column list or the hidden column options are specified.

IMPLICITLYHIDDENINCLUDE

The utilities assume that the implicitly hidden columns are included by default, when neither the column list nor the hidden column options are specified.

Consider the following examples of how the setting for **DB2_DMU_DEFAULT** affects the result of a load operation:

DB2_DMU_DEFAULT is set as IMPLICITLYHIDDENMISSING

db2 load from delfile1 of del insert into table1

If table1 has implicitly hidden columns, the load utility assumes that the data for implicitly hidden columns is not in the input file.

- **DB2_DMU_DEFAULT** is set as IMPLICITLYHIDDENINCLUDE

db2 load from delfile1 of del insert into table1

If table1 has implicitly hidden columns, the load utility assumes that the data for implicitly hidden columns is in the input file and attempts to load it.

DB2_DOCHOST

- Operating system: All
- Default: Not set (but Db2 will still try to access the Information Center from the IBM website), Values: http://hostname where hostname= valid host name or IP address
- Specifies the host name on which the *Db2 Information Center* is installed. This variable can be automatically set during the installation of the *Db2 Information Center* if the automatic configuration option is selected in the Db2 Setup wizard.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

- Operating system: All
- Default: NULL, Values: any valid port number
- Specifies the port number through which the Db2 help system serves the Db2 documentation. This variable can be automatically set during the installation of the *Db2 Information Center* if the automatic configuration option is selected in the Db2 Setup wizard.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2DSDRIVER_CFG_PATH

- Operating system: All
- Default: NULL
- This variable specifies multiple path and name pairs for the db2dsdriver.cfg configuration file. If you specify only a path for a path and name pair, the file name defaults to a value of db2dsdriver.cfg. If you specify only a file name for a pair, the path defaults to a location that is based on your operating system and driver type. You can specify any file extension, including .cfg. For details, see the db2dsdriver.cfg configuration file location.

A single path name pair can have a maximum of 255 characters. The maximum number of characters for the environment variable value depends on the operating system.

- Use the following delimiter characters to distinguish between each path and name pair:
 - On Windows operating systems, use the semicolon (;).
 - On Linux and UNIX operating systems, use either a semicolon (;) or a colon (:). You cannot use the semicolon character and the colon character together as a delimiter.

Do not use the delimiter character in the directory name.

- The directories are searched sequentially from left to right in the order that you specify for the value of the DB2DSDRIVER_CFG_PATH variable.
- The period (.) specifies the current directory.

DB2DSDRIVER_CLIENT_HOSTNAME

- Operating system: All
- Default: NULL
- Used to override the default client host name of the (db2dsdriver.cfg) configuration file. This variable forces CLI to pick the client host name entry from the automatic client reroute section of db2dsdriver.cfg file.

DB2_ENABLE_AUTOCONFIG_DEFAULT

- Operating system: All
- Default: YES, Values: YES or NO
- This variable controls whether the Configuration Advisor is run automatically at database creation. You do not need to restart the instance after you change this variable. If you execute the AUTOCONFIGURE command or run CREATE DB AUTOCONFIGURE, these commands override the setting of DB2_ENABLE_AUTOCONFIG_DEFAULT.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2_ENABLE_LDAP

- Operating system: All
- Default: NO, Values: YES or NO
- Specifies whether or not the Lightweight Directory Access Protocol (LDAP) is used. LDAP is an access method to directory services.

DB2_EVMON_EVENT_LIST_SIZE

- Operating system: All
- Default: 0 (no limit), Values: A value specified in KB/Kb/kb, MB/Mb/mb, or GB/Gb/gb; While there is no fixed upper limit for this variable, it is limited by the amount of available memory from the monitor heap.

Important: This registry variable is deprecated in Version 10.5 and might be removed in a future release. Setting this registry variable has no effect because of the changes that were made to the underlying event monitor infrastructure in Version 9.7.

- This registry variable specifies the maximum number of bytes that can be queued up waiting to be written to a particular event monitor. Once this limit is reached, agents attempting to send event monitor records will wait until the queue size drops below this threshold.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

Note: If activity records cannot be allocated from the monitor heap, they will be dropped. To prevent this from happening, set the **mon_heap_sz** configuration parameter to AUTOMATIC. If you have **mon_heap_sz** set to a specific value, ensure that **DB2_EVMON_EVENT_LIST_SIZE** is set to a smaller value. These actions, however, cannot guarantee that activity records will not be dropped, as the monitor heap is also used for tracking other monitor elements.

DB2_EVMON_STMT_FILTER

- Operating system: All
- Default: Not set; Values:
 - ALL: Indicates that the output for all statement event monitors is to be filtered. This option is exclusive.
 - 'nameA nameB nameC': Where each name in the string represents the name of an event monitor for which records are to be filtered. If more than one name is supplied, each name must be separated by a single blank. All input names will be made uppercase by Db2. The maximum number of event monitors you can specify is 128. Each monitor name can be up to a maximum of 128 characters long.
 - 'nameA:op1,op2 nameB:op1,op2 nameC:op1': Where each name in the string represents the name of an event monitor for which records are to be filtered. Each option (op1, op2, etc) represents an integer value mapping to a specific SQL operation. Specifying integer values allows users to determine which rules to apply to which event monitor.

• DB2_EVMON_STMT_FILTER can be used to reduce the number of records written by a statement event monitor. When set, this registry variable causes only the records for the following SQL operations to be written to the specified event monitor:

Table 6. Values to use for DB2_EVMON_STMT_FILTER to restrict event monitor output to specific SQL operations

SQL operation	Integer value mapping
EXECUTE	2
EXECUTE_IMMEDIATE	3
CLOSE	6
STATIC COMMIT	8
STATIC ROLLBACK	9

All other operations will not appear in the output of the statement event monitor. To customize the set of operations for which records are written to the event monitor, use integer values.

Example 1:

db2set DB2_EVMON_STMT_FILTER= 'mon1 monitor3'

In this example, mon1 and monitor3 event monitors will receive a record for a restricted list of application requests. For example, if an application being monitored by the mon1 statement event monitor prepares a dynamic SQL statement, opens a cursor based on that statement, fetches 10,000 rows from that cursor, and then issues a cursor close request, only a record for a close request will appear in the mon1 event monitor output.

Example 2:

db2set DB2_EVMON_STMT_FILTER='evmon1:3,8 evmon2:5,9'

In this example, evmon1 and evmon2 will receive a record for a restricted list of application requests. For example, if an application being monitored by the evmon1 statement event monitor issues a create statement , only the execute immediate and static commit operations will appear in the evmon1 event monitor output. If an application being monitored by the evmon2 statement event monitor performs SQL involving both a fetch and a static rollback only these two operations will appear in the evmon2 event monitor output.

Note: Refer to the sqlmon.h header file for definitions of database system monitor constants.

DB2_EXTSECURITY

- Operating system: Windows
- Default: YES, Values: YES or NO
- Prevents unauthorized access to Db2 by locking by locking Db2 objects (system files, directories, and IPC objects). To avoid potential problems, this registry variable should not be turned off. If DB2_EXTSECURITY is not set, its value is interpreted as YES on Db2 database sever products and N0 on clients.

DB2_FALLBACK

• Operating system: Windows

- Default: OFF, Values: ON or OFF
- This variable allows you to force all database connections off during the fallback processing. It is used in conjunction with the failover support in the Windows environment with Microsoft Cluster Server (MSCS). If **DB2_FALLBACK** is not set or is set to 0FF, and a database connection exists during the fall back, the Db2 resource cannot be brought offline. This will mean the fallback processing will fail.

DB2_FMP_COMM_HEAPSZ

- Operating system: Windows, UNIX
- Default: 20 MB, or enough space to run 10 fenced routines (whichever is larger).
- This variable specifies, in 4 KB pages, the size of the pool used for fenced routine invocations, such as stored procedure or user-defined function calls. The space used by each fenced routine is twice the value of the **aslheapsz** configuration parameter.

On UNIX, the underlying shared memory allocation is rounded up to a multiple of 256 MB by default, and the heap is allowed to grow into the pre-allocated shared memory as required. If the value is set explicitly, the underlying shared memory allocation is not rounded up, and the FMP heap is limited to the specified size. Pre-allocated shared memory does not count towards instance memory usage, but it does contribute towards the virtual memory consumption (swap on Solaris and HP-UX, configured virtual limits on Linux, Solaris, and HP-UX)

If you are running a large number of fenced routines on your system, you might need to increase the value of this variable. If you are running a very small number of fenced routines, you can reduce it to conserve virtual memory consumption.

Setting this value to 0 means that no set is created, and as a result no fenced routines can be invoked. It also means that the health monitor and the automatic database maintenance functionality (such as automatic backups, statistics collection, and REORG) will be disabled since this functionality relies on the fenced routine infrastructure.

If you are running SAS in-database analytics (enabled by setting the **DB2_SAS_SETTINGS** registry variable), the memory for connections to the SAS embedded process (EP) is also allocated from the FMP heap. Guidelines for fenced routines apply when the heap is adjusted to accommodate connections running queries that include in-database analytics. As a general rule, you can expect the FMP heap memory requirements to increase by 120 KB. If, however, you specify the COMM_BUFFER_SZ option for the **DB2_SAS_SETTINGS** registry variable, the FMP heap memory requirements increase by twice the value of the COMM_BUFFER_SZ option multiplied by the number of concurrent SAS queries that you want to support.

DB2_GRP_LOOKUP

- Operating system: Windows
- Default: NULL, Values: LOCAL, DOMAIN, TOKEN, TOKENLOCAL, TOKENDOMAIN
- This variable specifies which Windows security mechanism is used to enumerate the groups to which a user belongs.

DB2_HADR_BUF_SIZE

- Operating system: All
- Default: 2*logbufsz

• This variable specifies the standby log receiving buffer size in unit of log pages. If not set, Db2 will use two times the primary **logbufsz** configuration parameter value for the standby receiving buffer size. The maximum size that can be specified is 4 GB. This variable should be set in the standby instance. It is ignored by the primary database.

If HADR synchronization mode (the hadr_syncmode database configuration parameter) is set to ASYNC, during peer state, a slow standby might cause the send operation on the primary to stall and therefore block transaction processing on the primary. A larger than default log-receiving buffer can be configured on a standby database to allow it to hold more unprocessed log data. This may allow for brief periods where the primary generates log data faster than the standby can consume it, without blocking transaction processing at the primary.

Note: A larger log receiving buffer size can help absorb peak transaction loads on the primary database, but the buffer will still fill up if the average replay rate on the standby database is slower than the log rate on the primary database.

DB2_HADR_NO_IP_CHECK

- Operating system: All
- Default: OFF, Values: ON |OFF
- Specifies whether to bypass IP check for HADR connections
- This variable is primarily used in Network Address Translation (NAT) environments to bypass IP cross check for HADR connections. Use of this variable is not recommended in other environments because it weakens the sanity check of the HADR configuration. By default, configuration consistency for the local and remote host parameters is verified when an HADR connection is established. Host names are mapped to IP addresses for the cross check. Two checks are performed:
 - HADR_LOCAL_HOST parameter on primary = HADR_REMOTE_HOST parameter on standby
 - HADR_REMOTE_HOST parameter on primary = HADR_LOCAL_HOST parameter on standby

The connection will be closed if the check fails.

When this parameter is turned on, no IP check occurs.

DB2_HADR_PEER_WAIT_LIMIT

- Operating system: All
- Default: 0 (meaning no limit) Values: 0 to max unsigned 32 bit integer, inclusive
- When the DB2_HADR_PEER_WAIT_LIMIT registry variable is set, the HADR primary database will break out of peer state if logging on the primary database has been blocked for the specified number of seconds because of log replication to the standby. When this limit is reached, the primary database will break the connection to the standby database. If the peer window is disabled, the primary database will enter disconnected state and logging resumes. If the peer window is enabled, the primary database leaves disconnected peer state upon re-connection or peer window expiration. Logging resumes once the primary leaves disconnected peer state.

Note: If you set **DB2_HADR_PEER_WAIT_LIMIT**, use a minimum value of 10 to avoid triggering false alarms.

This parameter has no effect on a standby database, but it is recommended that the same value be used on both primary and standby databases. Invalid values (not a number or negative numbers) will be interpreted as 0, meaning no limit. This parameter is static. Database instance needs to be restarted to make this parameter effective.

DB2_HADR_REPLAY_ONLY_WINDOW_DIAGLEVEL

- Operating system: All
- Default: 1, Values: 0-2
- For databases that are configured with High Availability Disaster Recovery (HADR) and the Reads on Standby (ROS) feature, this variable controls the reporting of the replay-only window to the Db2 diagnostic log (db2diag.log). These diagnostic messages are useful for you to determine which operations on the Primary database are causing replay-only windows to be activated on the Standby database(s). More information about the replay-only window is available in Knowledge Center topic: **Replay-only window on the active standby database**. The following values are possible:
 - 0: Diagnostic messages only report the start and end of the replay-only window. There is no reporting of the operation that causes the activation of the replay-only window.
 - 1: Additional diagnostic messages report the operation that causes the replay-only window. Only the first operation in each replay only window is reported.
 - 2: Additional diagnostic messages report all the operations that can cause the replay-only window.
- When the LOG_DDL_STMTS database configuration parameter is set to YES, the DDL statement text is included in these diagnostic message(s).
- This variable is available in Version 11.1.3.3 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

DB2_HADR_ROS

- Operating system: All
- Default: OFF Values: OFF or ON
- This variable enables the HADR reads on standby feature. When **DB2_HADR_ROS** is enabled on the HADR standby database, the standby accepts client connections and allows read-only queries to run on it. **DB2_HADR_ROS** is a static registry variable, so it requires the Db2 instance to be restarted for a changed setting to take effect.

DB2_HADR_SORCVBUF

- Operating system: All
- Default: Operating system TCP socket receive buffer size, Values: 1024 to 4294967295
- This variable specifies the operating system (OS) TCP socket receive buffer size for the HADR connection, which allows users to customize the HADR TCP/IP behavior distinctly from other connections. Some operating systems will automatically round or silently cap the user specified value. The actual buffer size used for the HADR connection is logged in the **db2diag** log files. Consult your operating system network

tuning guide for the optimal setting for this parameter based on your network traffic. This variable should be used in conjunction with **DB2_HADR_SOSNDBUF**.

DB2_HADR_SOSNDBUF

- Operating system: All
- Default: Operating system TCP socket send buffer size, Values: 1024 to 4294967295
- This variable specifies the operating system (OS) TCP socket send buffer size for the HADR connection, which allows users to customize the HADR TCP/IP behavior distinctly from other connections. Some operating systems will automatically round or silently cap the user specified value. The actual buffer size used for the HADR connection is logged in the **db2diag** log files. Consult your operating system network tuning guide for the optimal setting for this parameter based on your network traffic. This variable should be used in conjunction with **DB2_HADR_SORCVBUF**.

DB2_HISTORY_FILTER

- Operating system: All
- Default: NULL, Values: NULL, G, L, Q, T, U
- This variable specifies operations that are not to modify the history file. You can use the **DB2_HISTORY_FILTER** registry variable to reduce potential contention on the history file by filtering out operations. Specify which operations that cannot modify the history file using a comma-separated list:

db2set DB2_HISTORY_FILTER=T, L

Possible values for DB2_HISTORY_FILTER are:

- **G**: Reorg operations
- L: Load operations
- **Q**: Quiesce operations
- **T**: Alter table space operations
- **U**: Unload operations

DB2_INDEX_PCTFREE_DEFAULT

- Operating system: All
- Default: Not set, Values: 0 to 99
- This registry variable specifies what percentage of each index page to leave as free space when building the index. The setting for **DB2_INDEX_PCTFREE_DEFAULT** is overridden if you explicitly specify the PCTFREE clause on the CREATE INDEX statement. The registry variable does not affect the LEVEL2 PCTFREE clause on the CREATE INDEX statement.

The registry variable does not apply at database upgrade time, even if the indexes are re-created during the upgrade. It only applies to a new installation or once the upgrade is complete. This registry variable is dynamic; you can set it or unset it without having to stop and start instance.

If $DB2_WORKLOAD$ is set to SAP, $DB2_INDEX_PCTFREE_DEFAULT$ will be set to 0.

DB2LDAP_BASEDN

• Operating system: All

- Default: NULL, Values: Any valid base distinguished domain name.
- When this is set, the LDAP objects for Db2 will be stored in the LDAP directory under
 - CN=System CN=IBM CN=DB2

under the base distinguished name specified.

When this is set for the Microsoft Active Directory Server, ensure that CN=DB2, CN=IBM, and CN=System are defined under this distinguished name.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2LDAPCACHE

- Operating system: All
- Default: YES, Values: YES or NO
- Specifies that the LDAP cache is to be enabled. This cache is used to catalog the database, node, and DCS directories on the local machine.

To ensure that you have the latest entries in the cache, do the following: REFRESH LDAP IMMEDIATE ALL

This command updates and removes incorrect entries from the database directory and the node directory.

DB2LDAP_CLIENT_PROVIDER

- Operating system: Windows
- Default: NULL (Microsoft, if available, is used; otherwise IBM is used.) Values: IBM or Microsoft
- When running in a Windows environment, Db2 supports using either Microsoft LDAP clients or IBM LDAP clients to access the LDAP directory. This registry variable is used to explicitly select the LDAP client to be used by Db2.

Note: To display the current value of this registry variable, use the **db2set** command:

db2set DB2LDAP_CLIENT_PROVIDER

DB2LDAPHOST

- Operating system: All
- Default:Null, Values: base_domain_name[:port_number], or
 base domain name:SSL:636 when using an SSL enabled LDAP host
- Specifies the host name and optional port number of the location for the LDAP directory where *base_domain_name* is the TCP/IP host name, and *[:port_number]* is the port number.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2LDAP_KEEP_CONNECTION

- Operating system: All
- Default: YES, Values: YES or NO

• Specifies whether Db2 caches its internal LDAP connection handles. When this variable is set to N0, Db2 will not cache its LDAP connection handles to the directory server. This will likely result in a negative performance impact, but it might be desirable to set

DB2LDAP_KEEP_CONNECTION to N0 if the number of simultaneously active LDAP client connections to the directory server needs to be minimized.

To maximize performance, this variable is set to YES by default.

The **DB2LDAP_KEEP_CONNECTION** registry variable is only implemented as a global level profile registry variable in LDAP, so you must set it by specifying the **-g1** option with the **db2set** command as follows: db2set **-g1** DB2LDAP KEEP CONNECTION=N0

DB2LDAP_SEARCH_SCOPE

- Operating system: All
- Default: DOMAIN, Values: LOCAL, DOMAIN, or GLOBAL
- Specifies the search scope for information found in database partitions or domains in the Lightweight Directory Access Protocol (LDAP). LOCAL disables searching in the LDAP directory. DOMAIN only searches in LDAP for the current directory partition. GLOBAL searches in LDAP in all directory partitions until the object is found.

DB2_LIMIT_FENCED_GROUP

- Operating system: Windows
- Default: NULL, Values: ON or OFF
- If you have Extended Security enabled, you can restrict the operating system's privileges of the fenced mode process (**db2fmp**) to the privileges assigned to the DB2USERS group by setting this registry variable to 0N and by adding the Db2 service account (the user name that runs the Db2 service) to the DB2USERS group.

Note: If LocalSystem is being used as the Db2 service account, setting **DB2_LIMIT_FENCED_GROUP** will have no effect.

You can grant additional operating system privileges to the **db2fmp** process by adding the Db2 service account to an operating system group that holds those additional privileges.

DB2_LOAD_COPY_NO_OVERRIDE

- Operating system: All
- Default: NONRECOVERABLE, Values: COPY YES or NONRECOVERABLE
- This variable will convert any LOAD COPY NO to either LOAD COPY YES or NONRECOVERABLE, depending on the value of the variable. This variable is applicable to HADR primary databases as well as to standard (non-HADR) databases; it is ignored on an HADR standby database. On an HADR primary database, if this variable is not set, LOAD COPY NO is converted to LOAD NONRECOVERABLE. The value of this variable either specifies a nonrecoverable load or the copy destination, using the same syntax as a COPY YES clause.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2LOADREC

- Operating system: All
- Default: NULL

- Used to override the location of the load copy during roll forward. If the user has changed the physical location of the load copy, **DB2LOADREC** must be set before issuing the roll forward.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2LOCK_TO_RB

- Operating system: All
- Default: NULL, Values: STATEMENT
- Specifies whether lock timeouts cause the entire transaction to be rolled back, or only the current statement. If **DB2LOCK_TO_RB** is set to STATEMENT, locked timeouts cause only the current statement to be rolled back. Any other setting results in transaction rollback.

DB2_MAP_XML_AS_CLOB_FOR_DLC

- Operating system: All
- Default: N0, Values: YES or N0
- The **DB2_MAP_XML_AS_CLOB_FOR_DLC** registry variable provides the ability to override the default DESCRIBE and FETCH behavior of XML values for clients (or DRDA Application Requestors) that do not support XML as a data type. The default value is N0, which specifies that for these clients a DESCRIBE of XML values will return BLOB(2GB), and a FETCH of XML values will result in an implicit XML serialization to BLOB that includes an XML declaration indicating an encoding of UTF-8.

When the value is YES, a DESCRIBE of XML values will return CLOB(2GB), and a FETCH of XML values will result in an implicit XML serialization to CLOB that does not contain an XML declaration.

Note: DB2_MAP_XML_AS_CLOB_FOR_DLC is deprecated and will be removed in a future release. This variable is no longer necessary because most existing Db2 applications that access XML values do so with an XML capable client.

DB2_MAX_LOB_BLOCK_SIZE

- Operating system: All
- Default: 0 (no limit), Values: 0 to 21487483647
- Sets the maximum amount of LOB or XML data to be returned in a block. This is not a hard maximum; if this maximum is reached on the server during data retrieval, the server finishes writing out the current row before generating a reply for the command, such as FETCH, to the client.

DB2_MEMORY_PROTECT

- Operating system: AIX with storage key support
- Default: NO, Values: NO or YES
- This registry variable enables a memory protection feature that uses storage keys to prevent data corruption in the buffer pool caused by invalid memory access. Memory protection works by identifying at which times the Db2 engine threads should have access to the buffer pool memory and at which times they should not have access. When **DB2_MEMORY_PROTECT** is set to YES, any time a Db2 engine thread tries to illegally access buffer pool memory, that engine thread traps.

Note: You will not be able to use the memory protection if **DB2_LGPAGE_BP** is set to YES. Even if **DB2_MEMORY_PROTECT** is set to YES, Db2 will fail to protect the buffer pool memory and disable the feature.

DB2_MIN_IDLE_RESOURCES

- Operating system: Linux
- Default: OFF, Values: OFF or ON
- This variable specifies that an activated database is to use minimal processing resources when it is idle. This might be useful in some virtual Linux environments (for example, zVM) where the small resource savings can help the host virtual machine monitor schedule its CPU and memory resources across all its virtual machines more efficiently.

DB2_NCHAR_SUPPORT

- Operating system: All
- Default: ON, Values: ON or OFF
- When this variable is set to ON (the default), the NCHAR, NVARCHAR and NCLOB spellings for the graphic data types are available for use in Unicode databases. Various national character related functions such as NCHAR() and TO_NCHAR() are also available.

This variable should only be set to OFF if an existing database has user defined types named NCHAR, NVARCHAR, or NCLOB.

Important: This registry variable is deprecated in Version 10.5 and might be removed in a future release. You no longer need to use user defined types called NCHAR, NVARCHAR or NCLOB because all the character length sensitive functions now return the number of characters. Start using the **nchar_mapping** database configuration parameter to specify the data type mapping for national character string data types in your Unicode databases before this registry variable is discontinued. For more details, see "nchar_mapping - National character mapping configuration parameter".

DB2N0EXITLIST

- Operating system: All
- Default: 0FF, Values: 0N or 0FF
- This variable indicates that Db2 should not load an exit list handler and that it should not perform a commit when the application exits, regardless of the setting of the DB2_COMMIT_ON_EXIT registry variable.

When **DB2NOEXITLIST** is turned off and **DB2_COMMIT_ON_EXIT** is turned on, any in-flight transactions for embedded SQL applications are automatically committed. It is recommended to explicitly add COMMIT or ROLLBACK statements when an application exits.

Applications that dynamically load and unload the Db2 library before the application terminates might crash when calling the Db2 exit handler. This crash might happen because the application attempts to call a function that does not exist in memory. To avoid this situation, set the **DB2NOEXITLIST** registry variable.

DB2_NUM_CKPW_DAEMONS

- Operating system: Linux and UNIX
- Default: 3, Values: 1[:FORK] to 100[:FORK]
- You can use the **DB2_NUM_CKPW_DAEMONS** registry variable to start a configurable number of check password daemons. The daemons are

created during **db2start** and handle check password requests when the default **IBMOSauthserver** security plugin is in use. Increasing the setting for **DB2_NUM_CKPW_DAEMONS** can decrease the time required to establish a database connection, but this is only effective in scenarios where many connections are being made simultaneously and where authentication is expensive.

DB2_NUM_CKPW_DAEMONS can be set to a value between 1 and 100. The database manager will create the number of daemons specified by **DB2_NUM_CKPW_DAEMONS**. Each daemon can handle check password requests directly.

An optional FORK parameter can be added to enable the check password daemons to explicitly spawn an external check password program (db2ckpw) to handle check password requests. This is similar to setting DB2_NUM_CKPW_DAEMONS to zero in previous releases. In FORK mode, each check password daemon will spawn the check password program for each request to check a password. The daemons in FORK mode are started as the instance owner.

If **DB2_NUM_CKPW_DAEMONS** is set to zero, the effective value is set to 3:FORK, where 3 check password daemons are started in FORK mode.

DB2_OPTSTATS_LOG

- Operating system: All
- Default: Not set (see details below), Values: OFF, ON {NUM | SIZE | NAME | DIR}
- DB2_OPTSTATS_LOG specifies the attributes of the statistics event logging files which are used to monitor and analyze statistics collection related activities. When DB2_OPTSTATS_LOG is not set or set to ON, statistics event logging is enabled, allowing you to monitor system performance and keep a history for better problem determination. Log records are written to the first log file until it is full. Subsequent records are written to the next available log file. If the maximum number of files is reached, the oldest log file will be overwritten with the new records. If system resource consumption is of great concern to you, disable this registry variable by setting it to OFF.
- Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

When statistics event logging is explicitly enabled (set to ON), there are a number of options you can modify:

- NUM: the maximum number of rotating log files. Default: 5, Values 1 15
- SIZE: the maximum size of rotating log files. (The size of each rotating file is SIZE/NUM.) Default = 15 Mb, Values 1 Mb 4096 Mb
- NAME: the base name for rotating log files. Default: db2optstats.number.log, for instance db2optstats.0.log, db2optstats.1.log, etc.
- DIR: the base directory for rotating log files. Default: diagpath/events

You can specify a value for as many of these options as you want, just ensure that ON is the first value when you want to enable statistics logging. For instance, to enable statistics logging with maximum of 6 log files, a maximum file size of 25 Mb, a base file name of mystatslog, and the directory mystats, issue the following command:

db2set DB2 OPTSTATS LOG=ON,NUM=6,SIZE=25,NAME=mystatslog,DIR=mystats

DB2REMOTEPREG

- Operating system: Windows
- Default: NULL, Values: Any valid Windows computer name
- Specifies the remote computer name that contains the Win32 registry list of Db2 instance profiles and Db2 instances. The value for **DB2REMOTEPREG** must be set only once after the Db2 database product is installed, and must not be changed after it is set. Use this variable with extreme caution.
- In a partitioned database environment, you can use the **DB2REMOTEPREG** registry variable to configure a computer that is not the instance owner to use the values of registry variables on the instance-owning computer. See "Setting variables at the instance level in a partitioned database environment" on page 6 for more information about when to use this variable.

When the Db2 database manager reads the registry variables on Windows operating systems, it reads the **DB2REMOTEPREG** value first. If the **DB2REMOTEPREG** variable is set, the database manager opens the registry on the remote computer that is specified by the **DB2REMOTEPREG** variable. Subsequent reading and updating of the registry variables is redirected to the specified remote computer.

For a computer that is not the instance owner to access the remote registry, the Remote Registry Service must be running on the target computer. Also, the user logon account and all Db2 service logon accounts must have sufficient access to the remote registry. To use the **DB2REMOTEPREG** variable, you must operate in a Windows domain environment so that you can grant the required registry access to the domain account.

• Do not use DB2REMOTEPREG in a Microsoft Cluster Server environment.

DB2_RESOLVE_CALL_CONFLICT

- Operating system: AIX, HP-UX, Solaris, Linux, Windows
- Default: YES, Values: YES, NO
- When routines called by triggers attempt to access tables that have been modified by other statements or routines in the body of the trigger, this can break nested SQL statement rules. Setting

DB2_RESOLVE_CALL_CONFLICT enforces that all modifications to tables are completed in compliance with the SQL standard rules for triggers before executing the CALL statement.

You must stop the instance before you reset **DB2_RESOLVE_CALL_CONFLICT** and then restart it. Then rebind any packages which cause invocation of triggers. To rebind SQL Procedures use: CALL

SYSPROC.REBIND_ROUTINE_PACKAGE

('P','procedureschema.procedurename','CONSERVATIVE');

You need to be aware that DB2_RESOLVE_CALL_CONFLICT can have a performance impact. Setting DB2_RESOLVE_CALL_CONFLICT to YES causes the Db2 database manager to resolve all potential read and write conflicts through the injection of temporary tables, as needed. Typically, the impact is small because at most one temporary table is injected. This has a small effect in an OLTP environment because only one (or a small number of) rows are being modified by the triggering statement. Typically, when following the general recommendation to use SMS (system managed space) for temporary table spaces, the performance impact from setting DB2_RESOLVE_CALL_CONFLICT is expected to be low.

Changes to this variable can take effect immediately for all future compiled SQL statements if the **db2set** command is issued with the **-immediate** parameter. You do not need to restart the instance.

DB2_RESTRICT_DDF

- Operating system: All
- Default: FALSE, Values: TRUE or FALSE
- Specifies whether the dynamic data format feature, also known as *progressive streaming* should be disabled. When **DB2_RESTRICT_DDF** is set to TRUE, the server informs the JDBC driver that the dynamic data format feature is to be disabled.

In SAP environments, when **DB2_WORKLOAD**=SAP is set, the default value of this registry variable is TRUE.

DB2ROUTINE_DEBUG

- Operating system: AIX and Windows
- Default: OFF, Values: ON or OFF
- Specifies whether to enable the debug capability for Java[™] stored procedures. If you are not debugging Java stored procedures, use the default, 0FF. There is a performance impact to enable debugging.

Note: DB2ROUTINE_DEBUG is deprecated and will be removed in a future release. This stored procedure debugger has been replaced by the Unified Debugger.

DB2_SAS_SETTINGS

- Operating system: All
- Default: Not set. Values: ENABLE_SAS_EP, LIBRARY_PATH, COMM_BUFFER_SZ, COMM_TIMEOUT, RESTART_RETRIES, DIAGPATH, DIAGLEVEL
- This variable is the primary point of configuration for in-database analytics with the SAS embedded process (EP). All options except for the ENABLE_SAS_EP option are configurable online.

ENABLE_SAS_EP

If you set this option to TRUE, the SAS EP starts automatically when you issue the **db2start** command. The default for this option is FALSE.

LIBRARY_PATH

The fully qualified path from which to load the SAS EP library the next time that the SAS EP process starts. If you do not specify a path, the Db2 database manager looks for the SAS EP library under the sqllib/function/sas directory. For security reasons, you should install the SAS EP library in a location where unauthorized users cannot modify or replace the file. Choose one of the following options:

- Ensure that the library path and the SAS EP library file are owned and can be written to only by the instance owner.
- Place the file in a directory, such as sqllib/function, that has the sticky bit set.

Only a user with SYSADM authority can configure the library path by using the **db2set** command.

COMM BUFFER SZ

An integer value specifying the amount of shared memory buffer, in

4 KB pages, to use for communication sessions between the Db2 data server and the SAS EP. The valid range of values for this parameter is 1 - 32767. The default value is 15. Communications buffers are allocated from the FMP communications heap. For more information, see DB2_FMP_COMM_HEAPSZ.

COMM_TIMEOUT

A timeout value that the Db2 database manager uses to determine whether the SAS EP should be considered unresponsive when exchanging control messages. If this value is reached, the database manager kills the SAS EP so that it can be spawned again. The default is 300 seconds.

RESTART_RETRIES

An integer value specifying the number of times that the Db2 database manager attempts to respawn the SAS EP after detecting that it has terminated abnormally. The valid range of values for this parameter is 0 - 100. The default value is 10. After the retry count has been reached, the database manager waits for 15 minutes before attempting the operation again.

DIAGPATH

A fully qualified path specifying the location of diagnostic logs for the SAS EP. The default value is the value of the **diagpath** database manager configuration parameter.

DIAGLEVEL

An integer value specifying the minimum severity level of messages that are captured in the SAS diagnostic logs. The valid values for this option are as follows:

- 1 Severe
- 2 Error
- 3 Warning
- 4 Informational

The default value is the value of the **diaglevel** database manager configuration parameter.

MEMSIZE

An integer value specifying the maximum amount of memory, in 4 KB pages, that the SAS EP can consume on a particular host. The valid range of values for this option is 1 - 4294967295. If there are multiple logical partitions, the value that is applied to each partition is divided by the number of logical partitions on the corresponding host. The default value is 20% of the value of the **instance_memory** database manager configuration parameter. If you set the **instance_memory** parameter to a fixed value, ensure that this value takes the additional memory requirements for the SAS EP into account.

Example:

db2set DB2_SAS_SETTINGS="ENABLE_SAS_EP:TRUE; LIBRARY_PATH:/home/instowner/sqllib/function/sas"

DB2SATELLITEID

• Operating system: All
- Default: NULL, Values: a valid satellite ID declared in the Satellite Control Database
- Specifies the satellite ID that is passed to the satellite control server when a satellite synchronizes. If a value is not specified for this variable, the logon ID is used as the satellite ID.

DB2_SERVER_CONTIMEOUT

- Operating system: All
- Default: 180, Values: 0 to 32767 seconds
- This registry variable and the DB2_DISPATCHER_PEEKTIMEOUT registry variable both configure the handling of a new client during connect time. DB2_SERVER_CONTIMEOUT allows you to adjust the time, in seconds, that an agent waits for a client's connection request before terminating the connection. In most cases, you should not need to adjust this registry variable, but if Db2 clients are constantly being timed out by the server at connect time, you can set a higher value for DB2_SERVER_CONTIMEOUT to extend the timeout period. If an invalid value is set, the default value is used. This registry variable is not dynamic.

DB2_SERVER_ENCALG

- Operating system: All
- Default: NULL, Values: AES_CMP or AES_ONLY
- •

Note: DB2_SERVER_ENCALG is deprecated in Version 9.7 and might be removed in a future release.

If the DB2_SERVER_ENCALG registry variable is set when you upgrade your instances to Db2 Version 9.7, the alternate_auth_enc configuration parameter is set to AES_ONLY or AES_CMP according to the setting of DB2_SERVER_ENCALG. Thereafter, to specify the encryption algorithm for encrypting user IDs and passwords, update the alternate_auth_enc configuration parameter. If the alternate_auth_enc configuration parameter is set, its value takes precedence over the DB2_SERVER_ENCALG registry variable value.

DB2SORT

- Operating system: All, server only
- Default: NULL
- This variable specifies the location of a library to be loaded at runtime by the load utility. The library contains the entry point for functions used in sorting indexing data. Use **DB2SORT** to exploit vendor-supplied sorting products for use with the load utility in generating table indexes. The path supplied must be relative to the database server.

DB2_STANDBY_ISO

- Operating system: All
- Default: NULL, Values: UR or OFF
- This variable coerces the isolation level requested by applications and statements running on an active HADR standby database to Uncommitted Read (UR). When DB2_STANDBY_ISO is set to UR, isolation levels higher than UR are coerced to UR with no warning returned. If the HADR standby takes over as the HADR primary, this variable will have no effect.

- Operating system: UNIX
- The registry variable controls a set of parameters which allow you to modify certain characteristics of the self tuning memory manager (STMM).
- Parameters:

GLOBAL_BENEFIT_SEGMENT_COMPATIBLE

- Default: Not set, Values: YES, NO
- The GLOBAL_BENEFIT_SEGMENT_COMPATIBLE parameter only has a functional impact if the database_memory configuration parameter is set to AUTOMATIC for a database.

This parameter influences the permission settings of the STMM shared memory segment. It should only be set to YES on systems with multiple instances, where some of the instances are downlevel and have **database_memory** set to AUTOMATIC, in order to mitigate downlevel compatibility issues that impact the tuning of a database's overall database memory usage. A downlevel instance would be an instance belonging to any of the following Db2 releases and fix pack levels: Db2 V9.1 at all fix pack levels, Db2 V9.5 fix pack 7 and earlier, and Db2 V9.7 fix pack 4 and earlier.

For instances that are non-root Db2 installations, you should set this variable only if you want all instances on the system make use of the same STMM shared memory segment. Leaving this variable unset or set to N0 will cause a non-root instance to use its own instance-specific STMM shared memory segment, which may impact the tuning of overall database memory usage for any databases with **database_memory** set to AUTOMATIC.

This registry variable is read once, during the Db2 instance startup. Note that you need to set this parameter across all the upgraded (that is, non-downlevel) instances and once set, you need to restart all upgraded instances.

GLOBAL_BENEFIT_SEGMENT_UNIQUE

- Default: Not set, Values: YES, NO
- The GLOBAL_BENEFIT_SEGMENT_UNIQUE parameter only has a functional impact if the database_memory configuration parameter is set to AUTOMATIC for a database.

This parameter specifies that each upgraded (that is, non-downlevel) instance is to make use of its own instance-specific STMM shared memory segment. The means that each instance tunes overall database memory usage for any of the databases belonging to it, independent of the tuning of overall database memory usage of databases belonging to the other instances on the system.

You should only consider setting this parameter to YES if the **instance_memory** configuration parameter is **not** set to AUTOMATIC for all instances on a system.

This registry variable is read once, during the Db2 instance startup. Note that this parameter needs to be set across all the upgraded instances and, once set, it requires that you restart all upgraded instances.

DB2_SYNC_RELEASE_LOCK_ATTRIBUTES

- Operating system: All
- Default: NO (YES when DB2_WORKLOAD=SAP), Values: YES, NO
- This variable specifies whether lock attributes that are used for lock avoidance can be synchronously removed during commit processing before any of the locks are released. This is to ensure a query (with currently committed or skip inserted feature) running during the processing of a commit would wait for these locks (instead of skipping the data) and be able to see all the data that is changed as part of the transaction that is committing.
- This variable affects row organized tables only.
- This variable is available in Version 11.1.3.3 and newer releases.
- Changes to this variable do not require the database instance to be restarted.

DB2_TRUNCATE_REUSESTORAGE

- Operating system: All
- Default: NULL (not set), Values: IMPORT, import
- You can use this variable to resolve lock contention between the **IMPORT** with **REPLACE** command and the **BACKUP** ... **ONLINE** command. In some situations, online backup and truncate operations are unable to execute concurrently. When this occurs, you can set **DB2_TRUNCATE_REUSESTORAGE** to IMPORT or import, and physical truncation of the object, including data, indexes, long fields, large objects and block maps (for multidimensional clustering tables), is skipped and only logical truncation is performed. That is, the **IMPORT** with **REPLACE** command empties the table, causing the object's logical size to decrease, but the storage on disk remains allocated.

This registry variable is dynamic; you can set it or unset it without having to stop and start instance. You can set

DB2_TRUNCATE_REUSESTORAGE before an online backup starts and then unset it after online backup completes. For multi-partitioned environments, the registry variable will only be active on the nodes on which the variable is set. **DB2_TRUNCATE_REUSESTORAGE** is only effective on DMS permanent objects.

In SAP environments, when **DB2_WORKLOAD**=SAP is set, the default value of this registry variable is IMPORT.

• Changes to this variable will take effect immediately for all future compiled SQL statements. There is no need to restart the instance or to issue the **db2set** command with the **-immediate** parameter.

DB2_UTIL_MSGPATH

- Operating system: All
- Default: *instanceName*/tmp directory
- The DB2_UTIL_MSGPATH registry variable is used in conjunction with the SYSPROC.ADMIN_CMD procedure, the SYSPROC.ADMIN_REMOVE_MSGS procedure, and the SYSPROC.ADMIN_GET_MSGS UDF. It applies on the instance level.
 DB2_UTIL_MSGPATH can be set to indicate a directory path on the server where the fenced user ID can read, write and delete files. This directory must be accessible from all coordinator partitions, and must have sufficient space to contain utility message files.

If this path is not set, the *instanceName*/tmp directory is used as the default (note that *instanceName*/tmp is cleaned up when Db2 is uninstalled).

If this path is not set when the ALTOBJ procedure is run, a temporary message file is created in the ~sqllib/tmp directory.

If this path is changed, the files that existed in the directory pointed to by the previous setting are not automatically moved or deleted. If you want to retrieve the contents of the messages created under the old path, you must manually move these messages (which are prefixed with the utility name and suffixed with the user ID) to the new directory to which DB2_UTIL_MSGPATH points. The next utility message file is created, read, and cleaned up in the new location.

The files under the DB2_UTIL_MSGPATH directory are utility specific, not transaction dependent. They are not part of the backup image. The files under the DB2_UTIL_MSGPATH directory are user managed; that means a user can delete the message files using the

SYSPROC.ADMIN_REMOVE_MSGS procedure. These files are not cleaned up by uninstalling Db2.

DB2_XBSA_LIBRARY

- Operating system: AIX, HP-UX, Solaris, and Windows
- Default: NULL, Values: Any valid path and file.
- Points to the vendor-supplied XBSA library. On AIX, the setting must include the shared object if it is not named shr.o. HP-UX, Solaris, and Windows do not require the shared object name. For example, to use Legato's NetWorker Business Suite Module for Db2, the registry variable must be set as follows:

db2set DB2_XSBA_LIBRARY="/usr/lib/libxdb2.a(bsashr10.o)"

The XBSA interface can be invoked through the **BACKUP DATABASE** or the **RESTORE DATABASE** commands. For example:

db2 backup db sample use XBSA

db2 restore db sample use XBSA

DB2_XSLT_ALLOWED_PATH

- Operating system: All
- Default: NULL or NONE, Values: ALL or a list of valid URIs , separated by a whitespace
- This registry variable controls how the Db2 instance refers to the external entities defined inside of an XSLT stylesheet.
 - NULL or NONE: No URI references are permitted, and the transformation with such a stylesheet fails.
 - ALL: All references to URIs are allowed.

Note: Uncontrolled reference to an external URI might be a severe security issue.

 List of URIs: Only references to URIs that are located in subdirectories of the URIs from the list are allowed, as shown in the following example:

db2set DB2_XSLT_ALLOWED_PATH = "http://some.website.com/test/dir /home/Joe/resource.txt"

Index

Α

aggregate registry variables 7

D

DB2_ALLOCATION_SIZE registry variable details 52 DB2_ALLOW_WLB_WITH_SEQUENCES registry variable details 36 DB2_ALTERNATE_GROUP_LOOKUP environment variable 22 DB2_ANTIJOIN registry variable details 46 DB2_ANTIJOIN variable 46 DB2_APM_PERFORMANCE variable 52 DB2_ATS_ENABLE registry variable details 74 DB2_AVOID_PREFETCH variable 52 DB2_BACKUP_USE_DIO registry variable details 52 DB2_BCKP_COMPRESSION registry variable details 74 DB2_BCKP_INCLUDE_LOGS_WARNING registry variable details 74 DB2_BCKP_PAGE_VERIFICATION registry variable details 74 DB2_CAPTURE_LOCKTIMEOUT registry variable details 11 DB2_CLI_DRIVER_INSTALL_PATH environment variable 22 DB2_CLP_EDITOR registry variable 41 DB2_CLP_HISTSIZE registry variable 41 DB2_CLPPROMPT registry variable 41 DB2_COLLECT_TS_REC_INFO registry variable 11 DB2_COMMIT_ON_EXIT registry variable 74 DB2_COMMON_APP_DATA_PATH registry variable 74 DB2_COMPATIBILITY_VECTOR registry variable overview 74 DB2_CONNRETRIES_INTERVAL registry variable details 11 DB2 COPY NAME environment variable 22 DB2_CPU_BINDING registry variable details 22 DB2_CREATE_DB_ON_PATH registry variable 74 DB2_DATABASE_CF_MEMORY 45

DB2_DDL_SOFT_INVAL registry variable details 74 DB2_DEFERRED_PREPARE_SEMANTICS registry variable details 46 DB2_DIAGPATH variable details 22 DB2_DISABLE_FLUSH_LOG registry variable 74 DB2 DISPATCHER PEEKTIMEOUT registry variable 74 DB2_DJ_INI variable 74 DB2_DMU_DEFAULT registry variable 74 DB2_DOCHOST variable 74 DB2_DOCPORT variable 74 DB2_ENABLE_AUTOCONFIG_DEFAULT variable 74 DB2_ENABLE_LDAP variable details 74 DB2_ENFORCE_MEMBER_SYNTAX registry variable 11 DB2_EVALUNCOMMITTED registry variable details 52 DB2_EVMON_EVENT_LIST_SIZE registry variable details 74 DB2_EVMON_STMT_FILTER registry variable details 74 DB2_EXPRESSION_RULES registry variable 11 DB2_EXTEND_COL_UNIQUE_INDEX_ACCESS details 11 registry variable 52 DB2_EXTENDED_IN2JOIN variable 52 DB2_EXTENDED_IO_FEATURES variable 52 DB2_EXTENDED_OPTIMIZATION variable 52 DB2_EXTSECURITY registry variable 74 DB2 FALLBACK variable 74 DB2_FCM_SETTINGS registry variable 43 DB2_FMP_COMM_HEAPSZ variable details 74 DB2_FORCE_APP_ON_MAX_LOG registry variable 11 DB2_FORCE_NLS_CACHE registry variable details 36 DB2_FORCE_OFFLINE_ADD_PARTITION registry variable 43 DB2_GRP_LOOKUP registry variable 74 DB2_HADR_BUF_SIZE variable 74 DB2_HADR_NO_IP_CHECK variable 74 DB2_HADR_PEER_WAIT_LIMIT registry variable 74 DB2_HADR_ROS registry variable 74 DB2_HADR_SORCVBUF registry variable 74

DB2_HADR_SOSNDBUF registry variable 74 DB2_HISTORY_FILTER registry variable details 74 DB2_INDEX_PCTFREE_DEFAULT registry variable details 74 DB2_INLIST_TO_NLJN registry variable 46 DB2_IO_PRIORITY_SETTING registry variable 52 DB2_KEEP_AS_AND_DMS_CONTAINERS_OPEN registry variable 52 DB2_KEEPTABLELOCK registry variable 52 DB2_LARGE_PAGE_MEM registry variable 52 DB2_LIC_STAT_SIZE registry variable 11 DB2_LIKE_VARCHAR registry variable details 46 DB2_LIMIT_FENCED_GROUP registry variable details 74 DB2_LOAD_COPY_NO_OVERRIDE variable 74 DB2_LOGGER_NON_BUFFERED_IO registry variable 52 DB2_MAP_XML_AS_CLOB_FOR_DLC registry variable details 74 DB2_MAX_CLIENT_CONNRETRIES registry variable DB2_MAX_GLOBAL_SNAPSHOT_SIZE registry variable 11 DB2_MAX_INACT_STMTS variable 52 DB2_MAX_LOB_BLOCK_SIZE variable 74 DB2_MAX_NON_TABLE_LOCKS variable 52 DB2_MCR_RECOVERY_PARALLELISM_CAP registry variable details 45 DB2_MDC_ROLLOUT registry variable 52 DB2_MEM_TUNING_RANGE variable 52 DB2_MEMORY_PROTECT registry variable 74 DB2_MIN_IDLE_RESOURCES registry variable details 74 DB2_MINIMIZE_LISTPREFETCH registry variable 46 DB2_MMAP_READ variable 52 DB2_MMAP_WRITE variable 52 DB2_NCHAR_SUPPORT registry variable details 74 DB2_NEW_CORR_SQ_FF variable 46

DB2_NO_FORK_CHECK registry variable details 52 DB2_NUM_CKPW_DAEMONS registry variable 74 DB2_NUM_FAILOVER_NODES registry variable 43 DB2_OBJECT_TABLE_ENTRIES registry variable 11 DB2_OPT_MAX_TEMP_SIZE registry variable 46 DB2_OPTSTATS_LOG registry variable 74 DB2_OVERRIDE_BPF variable 52 DB2_PARALLEL_IO registry variable overview 22 DB2_PARTITIONEDLOAD_DEFAULT registry variable 43 DB2_PINNED_BP registry variable 52 DB2_PMAP_COMPATIBILITY registry variable details 22 DB2_PMODEL_SETTINGS registry variable 36 DB2_RCT_FEATURES registry variable 52 DB2_REDUCED_OPTIMIZATION registry variable details 46 DB2_RESOLVE_CALL_CONFLICT variable 74 DB2_RESOURCE_POLICY registry variable 52 DB2_RESTORE_GRANT_ADMIN_AUTHORITHESriable 52 registry variable details 22 DB2_RESTRICT_DDF registry variable 74 DB2_SAS_SETTINGS registry variable details 74 DB2_SELECTIVITY registry variable 46 DB2_SELUDI_COMM_BUFFER registry variable 52 DB2_SERVER_CONTIMEOUT registry variable 74 DB2_SERVER_ENCALG registry variable details 74 DB2_SET_MAX_CONTAINER_SIZE registry variable 52 DB2_SKIPDELETED registry variable details 52 DB2_SKIPINSERTED registry variable details 52 DB2_SMP_INDEX_CREATE registry variable 52 DB2_SMS_TRUNC_TMPTABLE_THRESH variable 52 DB2_SORT_AFTER_TQ variable 52 DB2_SQLROUTINE_PREPOPTS registry variable details 46 DB2_SQLWORKSPACE_CACHE registry variable 52 DB2_SRVLSTLOG_LEVEL registry variable 11 DB2_STANDBY_ISO registry variable 74 DB2_SYSTEM_MONITOR_SETTINGS registry variable 11

DB2_TRUNCATE_REUSESTORAGE registry variable 74 DB2_TRUSTED_BINDIN registry variable 52 variable 22 DB2_USE_ALTERNATE_PAGE_CLEANING registry variable details 52 DB2_USE_FAST_LOG_PREALLOCATION registry variable 52 DB2_USE_FAST_PREALLOCATION registry variable 52 DB2_USE_IOCP registry variable 52 DB2_USE_PAGE_CONTAINER_TAG variable details 22 DB2_UTIL_MSGPATH registry variable 74 DB2_VIEW_REOPT_VALUES registry variable 11 DB2_WORKLOAD aggregate registry variable details 22 DB2_XBSA_LIBRARY registry variable 74 DB2_XSLT_ALLOWED_PATH registry variable 74 DB2ACCOUNT registry variable details 11 DB2ADMINSERVER variable 74 DB2ASSUMEUPDATE registry DB2AUTH registry variable 74 DB2BIDI registry variable details 11 DB2BPVARS registry variable 52 DB2BQTIME registry variable 41 DB2BQTRY registry variable 41 DB2CHECKCLIENTINTERVAL variable 36 DB2CHGPWD_EEE registry variable 43 DB2CHKPTR variable 52 DB2CHKSQLDA variable 52 DB2CLIINIPATH variable details 74 DB2CODEPAGE registry variable details 11 DB2COMM registry variable details 36 DB2CONNECT_DISCONNECT_ON_INTERRDB2NTWORKSET variable 52 variable 74 DB2CONNECT_ENABLE_EURO_CODEPAGE environment variable 22 DB2CONNECT_IN_APP_PROCESS environment variable 22 DB2CONSOLECP registry variable 11 DB2DBDFT variable 11 DB2DBMSADDR registry variable 22 DB2DISCOVERYTIME registry variable 11 DB2DOMAINLIST variable details 22 DB2DSDRIVER_CFG_PATH registry variable 74 DB2DSDRIVER_CLIENT_HOSTNAME registry variable 74

DB2ENVLIST environment variable 22 DB2FCMCOMM variable 36 DB2FODC registry variable details 11 DB2_UPDDBCFG_SINGLE_DBPARTITION DB2GRAPHICUNICODESERVER registry variable details 11 DB2INCLUDE registry variable 11 DB2INSTANCE environment variable details 22 DB2INSTDEF registry variable details 11 DB2INSTOWNER registry variable 11 DB2INSTPROF registry variable details 22 DB2IQTIME registry variable 41 DB2LDAP_BASEDN variable details 74 DB2LDAP_CLIENT_PROVIDER registry variable details 74 DB2LDAP_KEEP_CONNECTION registry variable details 74 DB2LDAP_SEARCH_SCOPE variable details 74 DB2LDAPCACHE variable 74 DB2LDAPHOST variable details 74 DB2LDAPSecurityConfig environment variable details 22 DB2LIBPATH environment variable 22 DB2LOADREC registry variable details 74 DB2LOCALE registry variable details 11 DB2LOCK_TO_RB variable 74 DB2LOGINRESTRICTIONS variable 22 DB2MAXFSCRSEARCH variable 52 DB2MEMDISCLAIM registry variable 52 DB2NODE environment variable details 22 DB2NOEXITLIST registry variable details 74 DB2NTMEMSIZE variable 52 DB2NTNOCACHE registry variable details 52 DB2NTPRICLASS registry variable 52 DB2OPTIONS environment variable details 22 DB2PATH environment variable 22 DB2PORTRANGE registry variable 43 DB2PRIORITIES registry variable 52 DB2PROCESSORS environment variable 22 DB2RCMD LEGACY MODE environment variable 22 DB2REMOTEPREG variable 74 DB2RESILIENCE environment variable details 22 DB2ROUTINE_DEBUG registry variable details 74 DB2RQTIME registry variable 41 DB2RSHCMD registry variable 36

DB2RSHTIMEOUT registry variable 36 DB2SATELLITEID variable 74 db2set command setting registry and environment variables 2 DB2SORCVBUF variable details 36 DB2SORT variable 74 DB2SOSNDBUF variable details 36 DB2STMM registry variable 74 DB2SYSTEM environment variable 22 DB2TCP_CLIENT_CONTIMEOUT registry variable 36 DB2TCP_CLIENT_KEEPALIVE_TIMEOUT registry variable details 36 DB2TCP_CLIENT_RCVTIMEOUT registry variable details 36 DB2TCP_SERVER_KEEPALIVE_TIMEOUT registry variable details 36 DB2TCPCONNMGRS registry variable 36 DB2TERRITORY registry variable details 11 DB2TMPDIR environment variable 22

Ε

environment variables overview 8 profile registries 1 setting Linux 5 partitioned database environment 6 process 2 UNIX 5 Windows 4

G

global-level profile registry 1

instance node-level profile registry 1 instance profile registry 1 instance-level profile registry overview 1 setting variables in partitioned database environment 6 instances current identifying 6 profile registry 1

Ρ

profile registries authorization requirements 2 environment variables 1 locations 2 profile registries (continued) registry variables 1

R

registry variables 11 aggregate 7 DB2_ALLOCATION_SIZE 52 DB2_ALTERNATE_GROUP_LOOKUP 22 DB2_ANTIJOIN 46 DB2_APM_PERFORMANCE 52 DB2_ATS_ENABLE 74 DB2_AVOID_PREFETCH 52 DB2_BACKUP_USE_DIO 52 DB2_BCKP_COMPRESSION 74 DB2_BCKP_INCLUDE_LOGS_WARNING DB2_BCKP_PAGE_VERIFICATION 74 DB2_CAPTURE_LOCKTIMEOUT 11 DB2_CLP_EDITOR 41 DB2_CLP_HISTSIZE 41 DB2_CLPPROMPT 41 DB2_COLLECT_TS_REC_INFO 11 DB2_COMMIT_ON_EXIT 74 DB2_COMMON_APP_DATA_PATH 74 DB2_COMPATIBILITY_VECTOR 74 DB2_CONNRETRIES_INTERVAL 11 DB2_COPY_NAME 22 DB2_CPU_BINDING 22 DB2_CREATE_DB_ON_PATH 74 DB2_DATABASE_CF_MEMORY 45 DB2_DDL_SOFT_INVAL 74 DB2_DEFERRED_PREPARE_SEMANTICS DB2_DIAGPATH 22 DB2_DISABLE_FLUSH_LOG 74 DB2_DISPATCHER_PEEKTIMEOUT 74 DB2_DJ_INI 74 DB2_DMU_DEFAULT 74 DB2_DOCHOST 74 DB2_DOCPORT 74 DB2_ENABLE_AUTOCONFIG_DEFAULT DB2_ENABLE_LDAP 74 DB2_ENFORCE_MEMBER_SYNTAX 11 DB2_EVALUNCOMMITTED 52 DB2_EVMON_EVENT_LIST_SIZE 74 DB2_EVMON_STMT_FILTER 74 DB2_EXPRESSION_RULES 11 DB2_EXTENDED_IO_FEATURES 52 DB2_EXTENDED_OPTIMIZATION 52 DB2_EXTSECURITY 74 DB2_FALLBACK 74 DB2_FCM_SETTINGS 43 DB2_FMP_COMM_HEAPSZ 74 DB2_FORCE_APP_ON_MAX_LOG 11 DB2_FORCE_NLS_CACHE 36 DB2_FORCE_OFFLINE_ADD_PARTITION DB2_GRP_LOOKUP 74 DB2_HADR_BUF_SIZE 74 DB2_HADR_NO_IP_CHECK 74 DB2_HADR_PEER_WAIT_LIMIT 74 DB2_HADR_ROS 74 DB2_HADR_SORCVBUF 74 DB2_HADR_SOSNDBUF 74 DB2_HISTORY_FILTER 74 DB2_INDEX_PCTFREE_DEFAULT 74 DB2_INLIST_TO_NLJN 46 DB2_IO_PRIORITY_SETTING 52

registry variables (continued) DB2_KEEP_AS_AND_DMS_CONTAINERS_OPEN 52 DB2_KEEPTABLELOCK 52 DB2_LARGE_PAGE_MEM 52 DB2_LIC_STAT_SIZE 11 DB2_LIKE_VARCHAR 46 DB2_LIMIT_FENCED_GROUP 74 DB2_LOAD_COPY_NO_OVERRIDE 74 DB2_LOGGER_NON_BUFFERED_IO 52 DB2_MAP_XML_AS_CLOB_FOR_DLC 74 DB2_MAX_CLIENT_CONNRETRIES 11 DB2_MAX_GLOBAL_SNAPSHOT_SIZE 11 DB2_MAX_INACT_STMTS 52 DB2_MAX_LOB_BLOCK_SIZE 74 DB2_MAX_NON_TABLE_LOCKS 52 7DB2_MCR_RECOVERY_PARALLELISM_CAP 45 DB2_MDC_ROLLOUT 52 DB2_MEM_TUNING_RANGE 52 DB2_MEMORY_PROTECT 74 DB2_MIN_IDLE_RESOURCES 74 DB2_MINIMIZE_LISTPREFETCH 46 DB2_MMAP_READ 52 DB2 MMAP WRITE 52 DB2_NCHAR_SUPPORT 74 DB2_NEW_CORR_SQ_FF 46 DB2_NO_FORK_CHECK 52 DB2_NUM_CKPW_DAEMONS 74 DB2_NUM_FAILOVER_NODES 43 DB2_OBJECT_TABLE_ENTRIES 11 DB2_OPT_MAX_TEMP_SIZE 46 DB2_OPTSTATS_LOG 74 DB2_OVERRIDE_BPF 52 46 DB2_PARALLEL_IO 22 DB2_PARTITIONEDLOAD_DEFAULT 43 DB2_PINNED_BP 52 DB2_PMAP_COMPATIBILITY 22 DB2_PMODEL_SETTINGS 36 DB2_RCT_FEATURES 52 DB2_REDUCED_ OPTIMIZATION 46 ⁷⁴ DB2_RESOLVE_CALL_CONFLICT 74 DB2_RESOURCE_POLICY details 52 DB2_RESTORE_GRANT_ADMIN_AUTHORITIES 22 DB2_RESTRICT_DDF 74 DB2_SAS_SETTINGS 74 DB2_SD_ALLOW_SLOW_NETWORK 45 DB2_SELECTIVITY 46 DB2_SELUDI_COMM_BUFFER 52 DB2_SERVER_CONTIMEOUT 74 DB2_SERVER_ENCALG 74 DB2_SET_MAX_CONTAINER_SIZE 52 DB2_SKIPDELETED 52 DB2_SKIPINSERTED 52 DB2_SMP_INDEX_CREATE 52 DB2_SMS_TRUNC_TMPTABLE_THRESH 52 DB2_SORT_AFTER_TQ 52 DB2_SQLROUTINE_PREPOPTS 46 DB2_SQLWORKSPACE_CACHE 52 DB2_SRVLSTLOG_LEVEL 11 DB2_STANDBY_ISO 74 DB2_SYSTEM_MONITOR_SETTINGS 11 DB2_TRUNCATE_REUSESTORAGE 74 DB2_TRUSTED_BINDIN 52 DB2_UPDDBCFG_SINGLE_DBPARTITION 22 DB2_USE_ALTERNATE_PAGE_CLEANING 52 DB2_USE_FAST_LOG_PREALLOCATION 52

registry variables (continued) registry variables (continued) DB2_USE_FAST_PREALLOCATION 52 DB2PORTRANGE 43 DB2_USE_IOCP 52 DB2PRIORITIES 52 DB2_USE_PAGE_CONTAINER_TAG 22 DB2PROCESSORS 22 DB2_UTIL_MSGPATH 74 DB2RCMD_LEGACY_MODE 22 DB2_VIEW_REOPT_VALUES 11 DB2REMOTEPREG 74 DB2_WORKLOAD DB2RESILIENCE 22 details 22 DB2ROUTINE_DEBUG 74 DB2_XBSA_LIBRARY 74 DB2RQTIME 41 DB2RSHCMD 36 DB2_XSLT_ALLOWED_PATH 74 DB2ACCOUNT 11 DB2RSHTIMEOUT 36 DB2ADMINSERVER 74 DB2SATELLITEID 74 DB2ASSUMEUPDATE 52 DB2SLOGON 11 DB2AUTH 74 DB2SORCVBUF 36 DB2BIDI 11 DB2SORT 74 DB2BPVARS 52 DB2SOSNDBUF 36 DB2STMM 74 DB2BQTIME 41 DB2SYSTEM 22 DB2BQTRY 41 DB2CHECKCLIENTINTERVAL 36 DB2TCP_CLIENT_CONTIMEOUT 36 DB2CHGPWD_ESE 43 DB2TCP_CLIENT_KEEPALIVE_TIMEOUT 36 DB2CHKPTR 52 DB2TCP_CLIENT_RCVTIMEOUT 36 DB2CHKSQLDA 52 DB2TCP_SERVER_KEEPALIVE_TIMEOUT 36 DB2CLIINIPATH 74 DB2TCPCONNMGRS 36 DB2CODEPAGE 11 DB2TERRITORY 11 DB2COMM 36 DB2TMPDIR 22 DB2CONNECT_DISCONNECT_ON_INTERR& 8 DB2CONNECT_ENABLE_EURO_CODEPAGE roza authorization requirements 2 DB2CONNECT_IN_APP_PROCESS 22 profile locations 2 DB2CONSOLECP 11 profile registries 1 DB2DBDFT 11 setting DB2DBMSADDR 22 partitioned database DB2DISCOVERYTIME 11 environments 6 DB2DOMAINLIST 22 procedure 2 DB2DSDRIVER_CFG_PATH 74 DB2DSDRIVER_CLIENT_HOSTNAME 74 DB2ENVLIST 22 U DB2FCMCOMM 36 user-level profile registry 1 DB2FODC 11 users DB2GRAPHICUNICODESERVER 11 profile registry 1 DB2INCLUDE 11 DB2INSTANCE 22 DB2INSTDEF 11 DB2INSTOWNER 11 DB2INSTPROF 22 DB2IQTIME 41 DB2LDAP_BASEDN 74 DB2LDAP_CLIENT_PROVIDER 74 DB2LDAP_KEEP_CONNECTION 74 DB2LDAP_SEARCH_SCOPE 74 DB2LDAPCACHE 74 DB2LDAPHOST 74 DB2LDAPSecurityConfig 22 DB2LIBPATH 22 DB2LOADREC 74 DB2LOCALE 11 DB2LOCK_TO_RB 74 DB2LOGINRESTRICTIONS 22 DB2MAXFSCRSEARCH 52 DB2MEMDISCLAIM 52 DB2NODE 22 DB2NOEXITLIST 74 DB2NTMEMSIZE 52 DB2NTNOCACHE 52 DB2NTPRICLASS 52 DB2NTWORKSET 52 DB2OPTIONS 22 DB2PATH 22

IBM.®

Printed in USA