New Features in IBM Informix Dynamic Server (IDS) Cheetah (IDS Version 11.10)

The following new features are introduced in the February 2007 Test Run of IBM IDS Cheetah (IBM Informix Dynamic Server Version 11.10). Watch the IDS Cheetah web site for download availability of future test run releases.
(http://www.ibm.com/software/data/informix/ids/new)

Theme 1: Scalability, High Availability and Performance

1. Multiple Remote Standalone Secondary Servers
2. Backup and Restore to Directories with Ontape
3. Continuous Logical Log Restore
4. Encrypted Communications for HDR
5. Improved Parallelism during Backup and Restore
6. RTO Policy to Manage Server Restart
7. Non-blocking Checkpoints
8. Performance Improvements for Enterprise Replication

Theme 2: Administration Free Zone

9. SQL Administration API
10. Schedule Administrative Tasks
11. Monitor and Analyze Recent SQL Statements
12. Dynamically Change Enterprise Replication Configuration Parameters and Environment Variables
13. Rename Enterprise Replication Columns, Tables, and Databases
14. Truncate Replicated Tables
15. Improved Statistics Maintenance
16. Installation Improvements on Windows Platforms
17. Session Configuration Routines

Theme 3: Integrated Solutions

18. Named Parameters in a JDBC Callable Statement
19. Indexable Binary Data Types
20. Trigger Enhancements
21. Derived tables in the FROM Clause of Queries
22. Index Self-Join Query Plans
23. Optimizer Directives in ANSI-Compliant Joined Queries
24. Deployment Wizard
25. Enhanced Concurrency with Committed Read Isolation
26. Enhanced Data Type Support for Cross-Server Distributed Queries
27. XML Publishing
28. Index Hierarchical Data
29. Basic Text Search
30. Improved Concurrency with Private Memory Caches for Virtual Processors

Feature Descriptions

Theme 1: Scalability, High Availability and Performance

1. Multiple Remote Standalone Secondary Servers

IDS Cheetah extends business continuity further with the introduction of the remote standalone secondary (RSS) feature. From a single HDR secondary, to multiple remote standalone secondary servers, there are now even more choices for disaster-recovery scenarios. RSS Servers can be geographically distant from the primary server, serving as remote back-up servers in disaster-recovery scenarios. Each RS Secondary Server maintains a complete copy of the database. Data integrity is maintained asynchronously over secure network connections. The RSS server is also available for active read access, allowing customers to maximize their investment and balance their workload. An RSS server is designed to co-exist with an HDR pair. The RSS server can be promoted to an HDR secondary in the event of a failure to the primary server.

RSS further extends the broad spectrum of IDS solutions for business continuity. This feature will be available for test run in March 2007. Watch the Cheetah web site for download availability. (http://www.ibm.com/software/data/informix/ids/new)

2. Backup and Restore to Directories with ontape

You can use the ontape utility to back up and restore data from a file system without interactive prompts. To enable this feature, set the TAPEDEV and LTAPEDEV configuration parameter to a valid directory name of a local or remote mounted file system. The ontape utility generates file names automatically and performs physical and log backups.

Ontape remains a popular alternative to ON-Bar for simple backup and restore operations. The ability to specify a directory with system generated filenames allows for easier backup and restore operations.

3. Continuous Logical Log Restore

This feature lets you perform continuous restore of logical log backups using the ontape and ON-Bar utilities. Use continuous log restore to keep a second system (hot backup) available to replace the primary system if the primary system fails. Logical logs backed

up on the primary system can be restored on the secondary system as they become available. If the primary system fails, the remaining available logical logs can be restored on the second system, which can then be brought online and function as the new primary system. You can suspend the log restore upon exit from the restore command and let the instance resume it with successive restore commands.

This feature provides a low-cost alternative to HDR where additional nodes can be set up to continuously restore logical logs from the Primary system. The difference between the IDS Cheetah release and earlier IDS versions is that the additional nodes are never brought to a quiescent state, thus not requiring operator intervention.

4. Encrypted Communications for HDR

You can encrypt communication between an HDR pair, to secure the transmission of data over unsecured networks, including the internet. You use new configuration parameters to enable encryption between the HDR servers and specify encryption options. After you enable encryption, the HDR primary database server encrypts the data before sending it to the secondary database server. The secondary database server decrypts the data. HDR encryption works in conjunction with Enterprise Replication encryption and operates whether Enterprise Replication encryption is enabled or not.

This feature ensures secure transmission between HDR pairs in addition to secure transmission already provided between IDS clients and servers. The feature uses the same OpenSSL libraries as other encryption facilities in IDS.

5. Improved Parallelism during Backup and Restore

This feature lets ON-Bar back up and restore a whole system using parallel I/O, which reduces the total time that is required to complete the backup or restore. Parallel whole system backups are still restorable to a consistent state without log backup and log restore.

This feature provides parallel physical (vs. logical) system backup via ON-Bar. The degree of parallelism is determined by the number of dbspaces in the customer environment. Issuing multiple streams of backup to ON-Bar will allow for parallel backup by the Storage Manager.

6. RTO Policy to Manage Server Restart

You can now create a recovery time objective (RTO) policy to set the amount of time, in seconds, that Informix Dynamic Server has to recover from a problem after you restart the server. You do this using a new configuration parameter, RTO_SERVER_RESTART. This configuration parameter enables failure recovery to meet the RTO policy by monitoring the workload and triggering checkpoints in a timely manner to ensure failure recovery will meet the policy.

1.1 New Features in IBM Informix Dynamic Server (IDS) Cheetah (IDS Version 11.10)

This feature dynamically adjusts to workload changes and maintains bufferpool flushing and checkpoint frequency.

7. Non-blocking Checkpoints

Informix Dynamic Server has replaced its checkpoint algorithm with a virtually non-blocking checkpoint algorithm. IDS now allows applications to continue to process transactions while checkpoint processing is occurring. IDS monitors the workload and past checkpoint performance and triggers checkpoints more frequently to avoid running out of critical resources, like the physical or logical log, to make sure transactions do not experience blocking during checkpoint processing. For applications that are sensitive to response times, the old method of using aggressive LRU flushing to reduce checkpoint quiescent times can be changed. LRU flushing can be less aggressive since transaction processing is not blocked during checkpoint processing. Less aggressive LRU flushing can improve transaction performance.

This feature eliminates the fuzzy checkpoint mechanism which can lead to certain undesirable transaction blocking intervals.

8. Performance Improvements for Enterprise Replication

Enterprise Replication has increased the degree of parallelism when applying transactions on target servers, resulting in better performance.

Theme 2. Administration Free Zone

9. SQL Administration API

A new SQL Administration API enables the DBSA to perform administrative tasks remotely by issuing SQL statements. The DBSA can now accomplish administrative tasks by invoking new built-in **admin( )** or **task( )** functions with arguments that emulate command-line arguments to the corresponding utility. For example, the following SQL statement, which is equivalent to the **oncheck -ce** command, instructs the database server to check the extents:

EXECUTE FUNCTION admin('check extents');

Some options can also accomplish tasks for which no corresponding utility exists. The effects of calling the admin( ) and task( ) functions with the same argument list are equivalent, but their return values, which indicate the result of the operation, have different data types. Information about the execution of Administration API functions is stored in the command_history table of a new sysadmin database of the Informix Dynamic Server instance.

The Administration API is open and published for use by any user interface program, such as a third party vendor that is interested in enhancing its administration tool for use with IDS.

10. Schedule Administrative Tasks

The new Scheduler allows you to manage and run scheduled maintenance, monitoring, and administration tasks at predefined times or as determined internally by the server. You can monitor activities (for example, space management) and create automatic corrective actions. Scheduler functions collect information and monitor and adjust the server, using an SQL-based administrative system and a set of tasks. A set of task properties, which define what needs to be collected or executed, control the Scheduler. The task properties are stored in the ph_task table in the sysadmin database. Each row in this table is a separate task and each column is a task property. You can modify task properties, and you can set up new tasks by inserting rows into the table.

This feature allows for common administration tasks to be scheduled by the DBA for automatic execution.

11. Monitor and Analyze Recent SQL Statements

You can now monitor the performance of recently executed SQL statements by configuring SQL statement tracing. This feature provides statistical information about each SQL statement executed on the system. The statistical information is stored in a circular buffer, which the DBA can resize. By default, this feature is turned off. The feature can be enabled for all users, or for just a specific set of users.

This feature allows for easy analysis of SQL statements for performance and tuning purposes.

12. Dynamically Change Enterprise Replication Configuration Parameters and Environment Variables

You can add, change, and remove in-memory values for Enterprise Replication configuration parameters and environment variables while the server is running. This feature does not update the onconfig file; the values are only valid for the current Enterprise Replication session. You use the following new commands to change values:

- cdr add config: to add a value
- cdr change config: to change an existing value
- cdr remove config: to remove an existing value

Note: the add and remove operations are not applicable to all Enterprise Replication configuration parameters and environment variables.

This feature allows for easier administration of Enterprise Replication nodes.

13. Dynamically Rename Enterprise Replication Columns, Tables, and Databases

You can now rename a replicated column, table, or database while Enterprise Replication is active. Use the RENAME statement to rename a column, table, or database on every participant in the replicate.

This feature allows for easier administration of Enterprise Replication nodes.

14. Truncate Replicated Tables

You can now use the TRUNCATE statement on replicated tables while replication is active. The truncate operation is useful prior to direct synchronization when target tables are significantly inconsistent.

15. Improved Statistics Maintenance

IDS now automatically collects index statistics, equivalent to the statistics gathered by UPDATE STATISTICS in LOW mode, when you create a B-tree index on a UDT column of an existing table or if you create a functional or VII index on a column of an existing table. Statistics that are collected automatically by this feature are available to the query optimizer, removing the necessity to manually run UPDATE STATISTICS. When B-tree indexes are created, column statistics are collected on the first index column, equivalent to what UPDATE STATISTICS generates in MEDIUM mode, with a resolution of 1% for tables of fewer than a million rows, and 0.5% for larger tables.

You can now view statistics about completed queries in the new Query statistics section in SET EXPLAIN.

In explicit UPDATE STATISTICS operations in MEDIUM mode, a new SAMPLING SIZE option in the Resolution clause can specify the minimum number of rows to sample for column distributions.

You can use the SET EXPLAIN statement to designate a non-default output file to capture the information that normally goes to the "sqexplain.out" file.

These features reduce the need for DBAs to perform UPDATE STATISTICS on indexes, enabling better query plans.

16. Installation Improvements on Windows Platforms

IDS on Windows now uses an industry-standard installation tool, which provides a shorter and easier installation requiring a substantially smaller footprint.

17. Session Configuration Routines

New built-in SPL procedures enable the Database Administrator to execute SQL and SPL statements automatically when a user connects to or disconnects from the database. These built-in procedures can be useful in setting the session environment or performing tasks like activating a role for users of applications whose code cannot easily be modified, or in automating operations that need to be performed after the application terminates.

1.1 New Features in IBM Informix Dynamic Server (IDS) Cheetah (IDS Version 11.10)

If the DBA specifies the login ID of a user as the owner of a procedure whose name is sysdbopen( ), IDS executes that procedure when the specified user connects to or disconnects from the database. If the DBA specifies PUBLIC as the owner, that routine is automatically executed when a user who is not the owner of any of these built-in session configuration procedures connects to the database. The sysdbopen( ) routine is not invoked, however, when a user who is already connected to a database performs a distributed operation, such as a cross-database or cross-server query, that references an object in another database.

Similarly, if no user.sysdbclose( ) is registered in the database for that user, another built-in procedure, user.sysdbclose( ) or public.sysdbclose( ) is called automatically when the user closes the connection to the database.

III. Theme 3: Integrated Solutions

18. Named Parameters in a JDBC CallableStatement

A CallableStatement provides a way to call a stored procedure on the server from a Java program. This feature provides support for named parameters in a CallableStatement, which was introduced in the JDBC 3.0 specification. Using named parameters in a CallableStatement adds the convenience of being able to identify parameters by name instead of by ordinal position. If the stored procedure is unique, you can omit parameters that have default values and you can enter the parameters in any order. Named parameters are especially useful for calling stored procedures that have many arguments when some of those arguments have default values.

19. Indexable Binary Data Types

The new Binary UDT DataBlade module provides two new data types that allow you to store binary-encoded strings, which can be indexed for quick retrieval. The binaryvar data type is a variable-length opaque type with a maximum length of 255 bytes. The binary18 data type is the same as the binaryvar data type except it holds a fixed value of 18 bytes. As part of a new DataBlade module, these data types come with string manipulation functions to validate the data types and bitwise operation functions that allow you to perform bitwise logical AND, OR, XOR, and NOT comparisons.

20. Trigger Enhancements

Several new features expand the syntax and the functionality of triggers on tables and on views:

- You can now define multiple INSERT, DELETE, UPDATE, and SELECT triggers on a table and multiple INSTEAD OF triggers for the view.
- When a table, view, or column list has multiple triggers for a DML event type, Informix Dynamic Server executes all BEFORE triggered actions before the FOR EACH ROW actions, and executes all FOR EACH ROW actions before the AFTER actions.
- You can create SPL procedures that refer to applicable OLD and NEW trigger correlated values. Within the procedure you can access applicable OLD and NEW values and modify the NEW values: e.g. using LET statements. From a FOR EACH ROW trigger action, you can execute this SPL procedure [syntax: execute procedure foo() with trigger references].
- New Boolean operators (DELETING, INSERTING, SELECTING, and UPDATING) can be used in procedures executed from trigger action statements. These test whether the currently executing triggered action was triggered by the

specified type of DML event and return a boolean value. The IF statement of SPL and the CASE expression of SQL can specify these operators as the condition in a trigger routine.

These features make it easier to incorporate IDS triggers on tables and on views within a heterogeneous information management system where multiple applications need to share the table or view.

You can find examples using multiple triggers in the demo directory: $INFORMIXDIR/demo/dbaccess/demo_ud/cr_trig.sql.

21. Derived tables in the FROM Clause of Queries

The SELECT statement can now include syntax that complies with ISO/IEC 9075:1992, the SQL-92 standard, to specify a full select subquery in the FROM clause as a data source for the query. These subqueries are called derived tables or table expressions, they can be simple, UNION, or joined subqueries, including OUTER joins, and can include the ORDER BY clause. In addition, AS correlation specifications in the FROM clause can declare temporary names for columns within the query. Informix-extension syntax, such as the FUNCTION keyword with iterator functions or the TABLE (MULTISET (SELECT ...)) keywords for collection-derived tables, can now be replaced in the FROM clause by SQL-92 syntax. This feature expands the capability of Informix Dynamic Server to run without modification queries that are interoperable on other database servers that support industry-standard SQL syntax.

You can find examples using derived tables in the demo directory: $INFORMIXDIR/demo/dbaccess/demo_ud/sel_sql99.sql.

22. Index Self-Join Query Plans

In earlier Informix Dynamic Server versions, queries of tables with composite indexes performed inefficiently if the ratio of duplicate values to the number of distinct values was much higher for the leading columns than for subsequent columns of the index. A new feature of the query optimizer supports a new type of index scan, called an index self-join path, that uses only subsets of the full range of a composite index. The table is logically joined to itself, and the more selective non-leading index keys are applied as index bound filters to each unique combination of the leading key values. By default, the optimizer considers this type of scan.

The optimizer also supports two new join-method directives, INDEX_SJ and AVOID_INDEX_SJ. The INDEX_SJ directive forces an index self-join path using the specified index, or choosing the least costly index in a list of indexes, even if data distribution statistics are not available for the leading index key columns. The AVOID_INDEX_SJ directive prevents a self-join path for the specified index or indexes. This feature can improve query performance on tables with composite indexes.

23. Optimizer Directives in ANSI-Compliant Joined Queries

Earlier IDS versions supported optimizer directives in Informix-extension joined queries, but not in queries that used ANSI/ISO syntax to specify joins. For both inline directives and external directives, this release extends support in ANSI/ISO joined queries to the following classes of optimizer directives:

- Access-method directives (FULL, AVOID_FULL, INDEX, AVOID_INDEX, INDEX_SJ, AVOID_INDEX_SJ)
- Explain-mode directives (EXPLAIN, AVOID_EXECUTE)
- Optimization-goal directives (ALL_ROWS, FIRST_ROWS).

The join-order directive (ORDERED) is supported only in ANSI/ISO-compliant LEFT OUTER joins and INNER joins. Because of ordering requirements for OUTER joins, in ANSI-compliant joined queries that specify the RIGHT OUTER JOIN or FULL OUTER JOIN keywords, the ORDERED join-order directive is ignored, but it is listed under Directives Not Followed in the sqexplain.out file.

This feature does the not support the join-method directives (USE_NL, AVOID_NL, USE_HASH, AVOID_HASH, /BUILD, and /PROBE) in ANSI/ISO joined queries, except in cases where the optimizer rewrites the query so that it is no longer uses the ANSI/ISO syntax.

24. Deployment Wizard

The new Deployment Wizard allows you to perform a custom installation of Informix Dynamic Server by selecting the components and features you want to install. This enables database administrators and independent software vendors to minimize the disk space (footprint) required for a custom installation of IDS. All installation methods (console, GUI, and silent) use the wizard to enforce dependencies between components and provide the estimated total footprint of selected components prior to actual file loading. You can selectively install or uninstall components at any time. You can use the generated response file to duplicate the installation setup in other installation locations.

25. Enhanced Concurrency with Committed Read Isolation

In Committed Read isolation level, exclusive row-level locks held by other sessions can cause SQL operations to fail when attempting to read data in the locked rows. This release introduces a new LAST COMMITTED keyword option to the SET ISOLATION COMMITTED READ statement to reduce the risk of locking conflicts when attempting to read a table. This new syntax instructs IDS to return the most recently committed version of the rows, even if another concurrent session holds an exclusive row-level lock.

This behavior can be extended to the Dirty Read, Read Uncommitted,

and Read Committed isolation levels by setting the new USELASTCOMMITTED configuration parameter or through new options to the SET ENVIRONMENT statement.

This feature supports B-tree indexes and functional indexes, but not R-tree indexes. It does not support tables that are being accessed by DataBlade modules, tables with columns of collection data types, tables created using a Virtual Table Interface, tables with page-level locking, tables with exclusive table-level locks, unlogged tables, or tables in databases with no transaction logging.

26. Enhanced Data Types Support in Cross-Server Distributed Queries

Enhanced Data-Type and UDR Support in Cross-Server Distributed Operations

Earlier releases of IDS restricted the remote execution of UDRs in databases of other IDS instances to SPL routines that the EXECUTE FUNCTION or EXECUTE PROCEDURE statement invoke explicitly, and to SPL routines that queries and other DML operations invoked implicitly.

This release extends support for UDRs in cross-database and cross-server distributed operations to most contexts where a UDR is valid in the local database. In addition, external routines written in the C or Java languages are now valid in any distributed operation where an SPL routine is valid.

This release also extends the data types that are valid as parameters or return values of cross-server UDRs, which were formerly restricted to non-opaque built-in SQL data types, by supporting these additional data types:

- BOOLEAN
- LVARCHAR
- DISTINCT of built-in types that are not opaque
- DISTINCT of BOOLEAN
- DISTINCT of LVARCHAR
- DISTINCT of the DISTINCT types listed above.

These data types can be returned by SPL, C, or Java language UDRs that use these data types as parameters or as return values, if the UDRs are defined in all the participating databases. Any implicit or explicit casts defined over these data types must be duplicated across all the participating Dynamic Server instances. The DISTINCT data types must have exactly the same data type hierarchy defined in all databases that participate in the distributed query.

This feature does not relax existing restrictions on other opaque and DISTINCT types or on large-object, serial, and collection data types in locally or remotely executed SPL routines or external routines.

27. XML Publishing

You can now perform XML publishing with Informix Dynamic Server. Built-in functions let you transform results of an SQL query to XML for use in XML applications or in a heterogeneous database environment. Other built-in functions let you use XPATH expressions to extract elements and values from XML documents.

28. Index Hierarchical Data

The node data type is part of the new Node DataBlade module which, with its supporting functions, gives you the ability to represent hierarchical data within the relational database. The advantage to this new data type is that it allows for searches within the hierarchy with a single SELECT statement using traditional operators without recursion. Represented as an ordinal number followed by either a single .0 or a set of ordinal numbers separated by dots, the node data type corresponds to a position in a tree structure similar to the way a table of contents represents chapter, section, and subsection information.

29. Basic Text Search

The Basic Text Search DataBlade module allows you to search words and phrases in an unstructured document repository stored in a column of a table. The column can be a CHAR, VARCHAR, LVARCHAR, TEXT, or CLOB data type. Search strategies include single and multiple character wildcard searches, fuzzy and proximity searches, and AND, OR and NOT Boolean operations. This feature is included with the database server at no extra cost.

30. Improved Concurrency with Private Memory Caches for Virtual Processors

You can now configure a private memory cache for every CPU virtual processor (CPUVP) to decrease the time of server memory allocation on large multiprocessor computers.