IBM Information Management

IBM® Informix® 12.10.xC1

New Features

Executive Summary

Powerful and simple have always been watchwords for the IBM® Informix® database server and version 12.10 continues in that vein with enhancements to five key feature areas – Data Warehousing, Sensor Data, Cloud, Embeddability, and Easy Administration. As well as updated platform support, there are also other new server features to improve SQL compatibility and boost performance.

Data Warehousing

New features in the Informix Warehouse Accelerator create a real time, in-memory data warehouse capability. An updated administration interface means that the accelerator is even easier to configure and manage on a daily basis by using the new automatic partition refresh and trickle feed features. Informix has never been faster, simpler to use or more consistent in its performance.

The Informix server has also been enhanced to provide powerful OLAP features, often used by Business Intelligence tools such as IBM Cognos® products, to simplify and speed up complex queries that have to work across very large sets of data.

Sensor Data

The ability to store and analyze high volumes of structured data is key to handling the current explosion of information from monitoring devices and Smart Meters. The unique capabilities of Informix spatial and TimeSeries data types make it a simple task to store and process this data avalanche. New enhancements built in to 12.10 make these data types even more powerful than before.

There is now a transparent view of time series data through Informix Warehouse Accelerator – marrying two Informix technologies into a very powerful data search and analysis engine. Management of time series data is now much simpler and faster with the capability to define active and dormant time ranges for the storage containers.

There are also several enhancements to the Virtual Table Interface and a new Loader API to make access to time series data even faster than before.

Spatial data support has been enhanced to include new features that come with the latest ESRI 10.1 library.

Cloud

New features in the Informix Flexible Grid environment make the configuration and management of sharded data within complex replication environments even simpler than before by introducing the powerful concept of regions and the easy configuration of spoke and hub systems.

Monitoring of replication queues has been enhanced to take account of their asynchronous nature and there are performance enhancements to allow for the replication of light-append operations.

The Connection Manager and internal server support for failover can now prioritize different networks within an application and database environment to maintain the highest level of service. This can also include keeping a Shared Disk Secondary cluster running while there is a network outage.

Embeddability

Autonomic capabilities have been extended with the ability to automatically compress data and

compression now supports more database objects, which lowers the storage space requirements. Dynamic configuration for more server parameters allows for automatic tuning and self healing, including the ability to manage parameters in groups using an export and import interface.

Rolling window definitions for fragmented tables boost performance and can be controlled automatically for simple administration.

Easy Administration

As well as boasting a new dashboard, the IBM OpenAdmin Tool for Informix continues to be developed to monitor and control many of the new Informix 12.10 features mentioned above, such as replication grids and Informix Warehouse Accelerator data marts. There is now a mobile app for DBAs on the move, allowing them to monitor the health and performance of all their database server systems.

The old Informix Storage Manager has been replaced with a new Primary Storage Manager to make the process of backing up the server simpler and faster.

Application Development Support

It is easier to build applications in general, and database neutral ones in particular, with the addition of several new SQL language elements. As well as the OLAP functions mentioned above there are new UNION operators, extensions for compatibility with other database servers (including Informix Parallel Server) and new aggregate capabilities.

Improvements to Informix Genero® speed up the development process with powerful new templates for the Business Application Modeling tool.

The Eclipse based loader tool for time series data now makes it easier to create load jobs, and has a simple command option for use in minimal production environments.

As always, there are a number of internal enhancements to the Informix server that improve cross-platform support and provide notable performance increases. Examples of performance enhancements include the following: ANSI style outer join queries can now be better optimized by utilizing hash joins and support for compression of data in the server has been extended to include indexes and simple large objects, which reduces I/O time.

"It is always the simple that produces the marvelous." - Amelia Barr

Contents

Data Warehousing

Informix Warehouse Accelerator

Informix Warehouse Accelerator was introduced in the 11.70.xC2 release and provided a snapshot view of an Informix database as an idealized, in-memory data mart. Using this feature, an Informix server is able to offload queries to the accelerator without the client being aware of a different data server being used. However, it was a static snapshot that would be invalidated if any updates were performed. While this was acceptable for historical data analysis, more real time data analysis would require regular snapshots to be retaken. In version 11.70.xC5, a manual update method was added to allow granular refresh at the partition level. This prevents having to reload the entire data mart but it was still a manual process. Two new features in 12.10 – Automatic Partition Refresh and Trickle Feed, allow the data mart to be kept up to date in near real time.

Automatic Partition Refresh

Partition Refresh works by keeping the data mart in step with the operational data by tracking active tables, which are also called partition fragments. If a fragment is dropped from the operational table then that data is also removed from the data mart. Similarly, when adding a fragment to a table, that data is transferred to the data mart. This requires a number of operations, which previously had to be done manually, to identify the partitions to be refreshed, the partitions to be dropped, and the partitions to be added.

The DBA had to perform the corresponding operation on the data mart for each partition. To refresh a partition containing data that has changed, the partition first needs to be dropped from the data mart and then loaded again, each as a separate step.

Even for a medium sized data mart containing several tables, some of them fragmented, identifying the partitions that need to be refreshed, added or dropped, can be a cumbersome task. While these operations can be scripted, the entire operation is non-trivial.

The metadata for tables that are part of a data mart and any partitioning of those tables is all stored in internal tables that the server can access. So, ideally, a single stored procedure can do all the work based on a few specified parameters. The purpose of the 12.10 feature automatic partition refresh is to provide such a stored procedure. This procedure can be scheduled for regular execution, either via Informix server's own task scheduler or via some external means, like an operating system **cron** job, without any further user intervention.

```
ifx_refreshMart ('accelerator name', 'data mart name', 'LOCKMODE');
```

The accelerator and data mart name parameters are self explanatory. The 'LOCKMODE' is one of:

- 'NULL' –The locking level that was specified the last time the `ifx_loadMart` function was used.

- 'TABLE' – Protects only the table that is being refreshed against changes. Each table is locked in share mode while it refreshes with data. The refreshed data is consistent within each table, but not necessarily across different tables.

- 'MART' – Protects all the tables that are referenced by the data mart against changes during the refresh operation. All the tables are locked in share mode until the data mart is refreshed. The refreshed data is consistent from all of the tables. All of the other user sessions are blocked from changing the data in the tables that are refreshing.

- 'NONE' – No locking. The data is read from the different tables without locking them. Other user sessions can change the data during the refresh operation. As a result, the refreshed data might be inconsistent. Data inconsistencies might be acceptable if the purpose of the data mart is to create statistics and discover trends, rather than to find the exact values of specific data rows.

The results of running the `ifx_refreshMart` function are like manually refreshing individual partitions. The `ifx_refreshMart` function will not change the definition of the data mart; it will always work only on existing tables in the data mart. Tables that have been newly added to the database are not considered for partition refresh because those tables are not part of the existing data mart definition. Likewise, dropping a table that is part of a data mart from the database will invalidate the complete data mart. The `ifx_refreshMart` function will not remove a table from a data mart's definition. To change the definition of a data mart, the existing data mart will need to be dropped and re-deployed with the new definition. In a high-availability cluster the `ifx_refreshMart` function can be run on the primary server or on the secondary server.

Trickle Feed

The Informix Warehouse Accelerator trickle feed feature is the ability for changes in the Informix database to be reflected in the accelerator data mart on a continuous basis rather than requiring a complete load or a partition update. For performance reasons there is a buffering process; so, trickle feed is considered to be a *near* real time operation. The time interval can be specified by the DBA.

A function, `ifx_setupTrickleFeed,` is executed once to set up trickle feed, specifying the accelerator name, data mart name, and the time interval for when buffers are flushed to the accelerator.

```
ifx_setupTrickleFeed('accelerator name', 'data mart name', buffer_time);
```

The buffer_time parameter is a positive or negative integer that represents both the time interval, in seconds, between refreshes, and indicates whether dimension tables are refreshed. The absolute value is used as the number of seconds between refreshes. For negative values, -1 through -8,639,999, only the fact tables are refreshed; dimension tables are not refreshed. For positive values, 1 through 8,639,999, both fact tables and dimensions are refreshed. There are 8,640,000 seconds in 100 days; the maximum interval is one second less. For best results it should be set to an absolute value of at least 10 seconds.

Inserts to the fact table will cause the rows to be captured and saved in buffers as an external table. At every specified time interval, Informix will push the buffered rows to the accelerator. Note that the accelerator compression dictionary is unchanged with trickle feed operations.

This operation is followed by a partition update operation on all dimension tables that have been modified during this interval. This invokes the `ifx_refreshMart` function described by the previous section. Upon completion, the accelerator instance is completely "in-sync" and ready for querying; all data has been inserted into the fact table and any updates to the dimension tables have been completed.

Note that partition update operations cannot be executed while trickle feed is enabled. A complete reload is recommended after extended trickle feed operations to refresh the accelerator compression dictionary with new data. This becomes mandatory after 32,000 cycles of the scheduled trickle feed task; the task will become idle at that point and will need to be restarted.

TimeSeries Integration

With Informix 12.10 it is now possible to load time series data into Informix Warehouse Accelerator directly, without materialization on disk. The virtual table, which is a relational view of the time series data, would normally be used by a business intelligence (BI) client, but the client would be unaware that the virtual table is optimized for access based on the time dimension. Using the virtual table like a fact table can now be handled in the accelerator using natural BI join methodologies. As with all accelerator operations this is completely transparent to the client.

Administering Informix Warehouse Accelerator

The first release of Informix Warehouse Accelerator required IBM Smart Analytics Optimizer Studio (ISAO) to perform administration tasks. The ISAO tool was bundled with the Informix Warehouse edition and required a separate installation to sync and work with the accelerator instance.

The 12.10 release provides a simple dashboard in OAT for Informix users to administer and monitor their warehouse contents. [Refer below to the section Administering Informix Warehouse Accelerator using OAT].

UNION Query Support

The SQL UNION query allows the combination of the result sets of two or more SQL SELECT statements. It can remove duplicate rows between the various SELECT statements, or, to keep all duplicate rows, specify UNION ALL between the SELECT statements.

The SQL UNION keyword is now supported by Informix Warehouse Accelerator as well, so each of the SELECT statements in a UNION query can be accelerated. A UNION query also provides the ability to combine results not only from two different tables, but also from tables across multiple data marts in a given instance, or multiple accelerator instances.

OLAP – On Line Analytical Processing

OLAP functions greatly simplify the writing of common business analytical queries such as running totals and ranking. The Informix implementation of OLAP is not only efficient syntactically; it has also been optimized for performance. The demand on database resources is greatly reduced compared to non-OLAP queries because the server does not have to spend time aggregating and re-aggregating over potentially millions of rows to compute the required OLAP functions.

Informix 12.10 introduces support for OLAP functions to provide ordinal ranking, row numbering, and aggregate information for subsets of the qualifying rows that a query returns. These subsets are based on partitioning the data by one or more column values.

OLAP functions can be used to define moving windows within a partition for examining dimensions of the data, and identifying patterns, trends, and exceptions within data sets. If one of the built-in aggregate functions listed below is invoked from an OLAP window, the return value is based on only the records within that window, rather than on the entire result set of the query. In nested queries, the set of rows to which each OLAP function or aggregate is applied is the result set of the query block that includes the OLAP function. There are various kinds of OLAP functions:

- Ranking functions
    - `RANK, DENSE_RANK (DENSERANK)`
    - `PERCENT_RANK, CUME_DIST`
    - `LEAD, LAG, NTILE`
- Numbering functions
    - `ROW_NUMBER (ROWNUMBER)`
- Aggregate functions
    - `SUM, COUNT, AVG, MIN, MAX`
    - `STDEV, VARIANCE, RANGE`
    - `FIRST_VALUE, LAST_VALUE`
    - `RATIO_TO_REPORT (RATIOTOREPORT)`

These OLAP functions are now also supported by Informix Warehouse Accelerator so that complex

BI queries can be further optimized for high performance.

Sensor Data

Informix has a very strong foothold in this area with two specialized sets of data types specifically designed for high volume, structured data: spatial, for data associated with a geographic location, and TimeSeries, for data that always has a time dimension. Because of the way these non-relational data types are uniquely integrated into the standard Informix server, they can be combined to provide a very efficient storage and analytics engine.

New TimeSeries Features

Being able to store and handle very large amounts of data efficiently is paramount for the Informix TimeSeries solution as it is part of the IBM portfolio of products that can be called upon to handle Big Data. For example, without the TimeSeries solution, it wouldn't be possible to manage the trillions of rows of data that can be found in some of the Energy and Utility company database systems. All the new TimeSeries features are targeted at improving the performance and manageability of such Big Data systems.

TimeSeries Plug-in for Data Studio

The Informix TimeSeries plug-in can be installed either through IBM Data Studio or IBM Optim Development Studio. The plug-in simplifies time series handling and is designed in such a way that it can be integrated into other Eclipse-based tools.

The plug-in allows the loading of time stamped data from a file, or an external database, into existing time series instances by creating load jobs. It is very simple to create a load job using the Eclipse tooling and existing job definitions can be easily reused to create new ones. After creating a load job, the command-line loader application can be used to run the jobs. Running load jobs from the command line needs only the Eclipse Platform Runtime Binary (minimal Eclipse platform) instead of requiring a full installation of Eclipse or IBM Data Studio. Informix server job scheduling facilitates the running time series load jobs at regular intervals in an unattended manner.

It is possible to load time series data in parallel by running multiple loader applications simultaneously. As the data is loaded, statistics about the load process are printed to the console. When the load job is complete, cumulative statistics for the whole run are printed. The maximum number of load jobs that can be run simultaneously is limited only by the capabilities of the server machine.

The usual mode of operation is to read the data from a file. However, it is also possible to load time series data directly from another database without having to export the data into a file. When creating a table definition, specify a connection to a database and a query to return the data that is to be loaded. It is possible to preview the returned data to validate the query and also set other properties of the load job within the plug-in. The external database source can be any database that supports JDBC for client connectivity.

TimeSeries Fast Loader API

The TimeSeries plug-in uses a low-level server interface to directly access data stored in the time series containers. In version 12.10, that interface has now been made publicly available so that a user can define their own fast loader for time series data. The key to providing such a high speed interface is to avoid standard SQL methods and use a function-based interface to send data to the server.

The first thing to do is to use the `TSL_Init` function to initialize the required global segments within the server. This needs to be done once for each table and TimeSeries column combination that is to be loaded and is referenced by a handle made up of the table and column name - 'table|column'. Each global segment remains in place until the server shuts down or the

`TSL_Shutdown` function is called for that handle.

Once the global segment is initialized, any number of parallel sessions can make use of it by using the `TSL_Attach` function. To insert the data, each loader session uses the `TSL_Put` function to send the specially formatted time series data to the server where it is stored in an internal buffer. Variations of the `TSL_Put` function are available that send data as an LVARCHAR character buffer, a CLOB, a ROW type, or by using `TSL_PutSQL` to specify a SELECT statement that is run within the server context to transfer rows from a staging table into the buffer.

To write the server's time series buffer to disk, the `TSL_Flush` function is used. The flush must be done within a transaction to preserve data integrity. Each loader session would call `TSL_Put` a number of times, flush the data, and then loop around again. The amount of data being sent should be controlled so that there is no danger of having so much work to do in the flush operation that a long transaction might be triggered. At the end of the load process, the session should call `TSL_SessionClose` to release any server resources it has allocated.

Increased Performance

A number of internal enhancements have been introduced with the TimeSeries solution that comes with Informix version 12.10. Most of these will be transparent to the user but some require application or database changes to be enabled.

Improved Concurrency

The internal storage mechanism for time series is through a database partition called a container. Within each container is a sub-set of the time series data with access controlled through the top levels of a B-tree index. The original design of the TimeSeries solution used a page-level lock to control access to the internal data structures. This would have a negative impact on performance unless steps were taken to separate out threads of execution between different containers – by using the `GetContainerName` function, for example.

The version 12.10 release now increases concurrent access to containers by using row level locking on the index nodes. This allows multiple writers to access the same container simultaneously but will increase the number of locks required – it may be necessary to adjust the server's LOCKS parameter. If there already is a partitioning scheme that can be used to naturally separate active writing sessions, then a new function, `TSContainerLock`, can be used to lock the whole container. This will increase performance by not having to have the insert process go through the lock manager.

Reduced Logging

If time-series elements are loaded into containers in a single transaction, the efficiency can be increased by specifying a reduced amount of logging. By default, every time-series element that is inserted generates two log records, one for the inserted element and one for the TimeSeries header update. This feature introduces a deferred approach to updating the header slot when performing inserts. Rather than update the header for every inserted element on a page, the header will only be updated at the end of the transaction.

For example, loading a days worth of data from a typical Smart Meter, with a 15 minute interval between readings, will generate 96 new elements. The combined insert plus header update will generate at least 192 log records. When using the reduced logging feature and doing the same set of inserts (within a transaction) only one header update is required. This practically halves the number of log records required.

The reduced logging feature can be turned on for inserts through a virtual table by using the TS_VTI_REDUCED_LOG flag when creating the virtual table. When using the TimeSeries Fast Loader API, as detailed above, the TSOPEN_REDUCED_LOG flag provides the same capability.

The flag value in both cases is 256. When using this feature it is not possible to select data from any time series instances that were modified until the transaction is complete.

Faster Virtual Table Access

Ordinarily an insert of a value through the a virtual table that is based on a TimeSeries column is translated into the equivalent `PutElem` method for the time series. Although functional, it is relatively slow to process data on such a row by-row basis. If the virtual table is created with the TS_VTI_ELEM_INSERT flag (value 128), then a faster interface is used when inserting multiple rows of data for the same time series. When using this method the following constraints must be met:

- The time series must use containers as the storage type.

- The base table must have a primary key.

- The time series instance must already exist in the table.

- No updates are performed on the base table data.

The query performance for virtual tables has also been improved by allowing the optimizer to push down WHERE clause predicates that include complex functions. This allows more efficient processing at the TimeSeries level, rather than having to transform all the data into tuples for processing in the SQL layer.

Rolling Window Containers

It is important to be able to manage very large amounts of data efficiently and one important task is the rolling in and out of data based on an aging criteria. This has been addressed for many years with standard Informix tables using expression based fragmentation with the attach and detach operators. Previously a manual task, this operation is enhanced in this Informix release with the introduction of automatic rolling interval fragments. [Refer to the section below which discusses Embeddability].

Time series data has traditionally been kept for indefinite amounts of time but the current explosion of Big Data, requiring the storage of very large amounts of operational information, means that extra thought needs to be given to a similar roll-in and roll-out of time series data. This requirement has been addressed with the rolling window container feature.

To simplify the administration of such a system the definition of a time series container has been extended to include a windowed container that allows the user to specify the way the data is handled with respect to time stamp age. Once the windowed container is defined and specified for a time series, then all operations required to manage the obsolescence of data within that time series can be made fully automatic.

The granularity of the window can be defined as day, week, month, or year with a set number of such windows being active at any one time. When data is inserted that creates a new active partition then the oldest active partition has to be moved. When such a partition is removed from the active window, it moves to the dormant window. The number of dormant partitions is also user definable.

For example consider a rolling window definition with 12 months active and 4 months dormant.

Data from September 2011 through to December 2012:
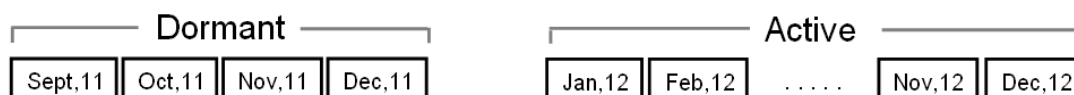
Data for January 2013 is added:

Data for January 2013 will be inserted into the new window but the data for September 2011 has been deleted. Administration commands allow the manual movement of data between the dormant and active windows; otherwise, the system can be totally automatic.

The management of the rolling window feature is accomplished by storing the partition information as a time series in its own right, linked to the definition of the windowed partition. The `TSContainerUsage` functions have been enhanced with a rolling window flag that can separate out the data usage between the active and dormant windows and the partition management information. Two new system tables, `TSContainerUsageActiveWindowVTI` and `TSContainerUsageDormantWindowVTI,` give full information on the state of each container window.

Replication of Time Series Data

Version 12.10 now supports replication of time series data through two of the more common Informix replication mechanisms: High-Availability Data Replication (HDR) and Enterprise Replication.

When using HDR, time series data will be automatically replicated from a primary server to the secondary server through the standard log replay mechanism. The time series data will be available for read-only access on that secondary server. Updatable HDR secondaries are not supported. Note that some TimeSeries objects, such as calendar and container definitions, are cached in memory. Care should be taken if these objects are modified without arranging for the definitions to be re-synced to the secondary server. However, it would be unusual to change these definitions, as it amounts to a change in the underlying TimeSeries data schema and is unlikely to happen on a regular basis.

When using Enterprise Replication, it is necessary to set the CDR_TSINSTANCEID `onconfig` parameter to a different number greater than zero for each instance that will be involved in replicating time series data. This ensures that the time series instance identifiers remain unique across all instances that are participating in time series replication. The containers and calendars used must also be declared identically across all instances. Containers must be of standard type; it is not possible to use automatically created or rolling window containers.

Spatial Data

Support for ESRI 10.1

The Informix Spatial solution is tailored to ESRI's ArcSDE application server and is based on ESRI's geometry engine (shape library). The Spatial solution now includes features that are available in the new ESRI SDE 10.1 libraries. There are new spatial functions that allow overlapping IDs for spatial referencing definitions, support for annotation, SDE format, and others. The integration provides consistent results of spatial operations on all software tiers: database, middleware (ArcSDE), and client (ArcGIS, ArcIMS). Support for the latest 10.x libraries eases the porting and migration efforts from other databases to Informix.

Cloud

The Informix Flexible Grid was introduced in version 11.50 as a means of creating a heterogeneous database server environment suitable for enterprise-ready cloud deployment. New features have been added to increase performance and simplify administration.

Managing Sharded Data

Queries across Sharded Data Systems

Informix 12.10 brings in a new concept called region, which defines a subset of nodes within a grid. The mapping of nodes to region is completely flexible, with no restriction on the number of regions within a grid and any node can be a member of multiple regions.

It is now possible to define multi-node queries that automatically convert a simple query into a UNION or UNION ALL query which spans the nodes within the grid or subset of the nodes within the grid. The target nodes of such a query are defined to be a grid, or a region within a grid.

- **Ignore unavailable grid nodes -** Avoid failure of a query if one or more nodes are unavailable by running the SET ENVIRONMENT GRID_NODE_SKIP ON statement before issuing the query. The identity of a skipped grid server can be found by executing the new `ifx_gridquery_skipped_nodes` SQL function.

- **Define tables for queries** - The **cdr change gridtable** command adds or removes grid tables.

- **Work with ALTER** - When an ALTER command is executed, the grid table metadata is flagged to indicate that a alter operation is in progress. The command **cdr remaster gridtable** can be used to verify that the alter has been successfully propagated throughout the grid.

Defer DDL Propagation

Defer the DDL propagation to a later time by queuing up the DDL changes, as part of the grid operation. This allows testing an application with new DDL changes before rolling out the changes the rest of the grid.

Propagate External Files Through a Grid

Informix 12.10 allows the replication of non-database objects such as load files, executable programs, scripts, UDR libraries by having the external files registered with Enterprise Replication. These files then get propagated along with the changes to all the nodes within a grid.

This replication of the object is performed by executing a new SQL procedure `ifx_grid_copy`, which takes the grid name and the path name of the file to be sent. The procedure makes it easy to replicate non-database objects in conjunction with the associated DDL changes. It is necessary to set an *onconfig* parameter, GRIDCOPY_DIR, to define the place where the files will be transferred. The execution of `ifx_grid_copy` must be done by an authorized grid user and from an authorized grid source node.

Replicates are Mastered by Default

By default, Enterprise Replication replicates are now master replicates. If a master server is not explicitly specified with the **--master** option, the first participant becomes the master replicate. In order to create a classic replicate, which does not verify the schemas of participants, the **--classic**

option must be included in the **cdr define replicate** command.

A master replicate uses saved dictionary information about the attributes of replicated columns to verify that participants conform to the specified schema. When the **cdr define replicate** or **cdr change replicate** command is run, all participants specified within that command are verified. If the schema does not need to be verified, create a classic replicate.

If any participant fails the verification, the command fails and that local participant is disabled. If a specified participant does not contain the master replicate table, Enterprise Replication automatically creates the table on the participant, based on the master replicate dictionary information. All database servers that have master replicates must be able to establish a direct connection with the master replicate database server.

Simplified set up of a Consolidation System

Typical consolidation models involve hub and spoke nodes where the spokes are remote systems which are replicating into a central server. The prior release of IBM Informix supported a receive-only target for replication and a primary. The primary member of a replicate definition is the source node for replication. However, the source node will also receive and apply data changes made on other source nodes. That makes it difficult to create a true consolidation model without having to define multiple replicates.

In the above scenario, it is necessary to define individual replicates for each spoke to hub table that is being replicated. This not only creates undo complexity in the definition of replicates, but also causes performance-related problems because each additional replicate increases the length of the cache chains that the server must traverse to find the attributes of an individual replicate.

Informix 12.10 introduces a feature that adds a new type of member to the replicate definition. This new member is defined to be a send-only participant. The server defined as send-only will only forward the replicated changes to the target nodes that are not send-only. Making the spoke nodes of a consolidation model into send-only nodes significantly reduces the replicate definitions, as all nodes can be now configured in a single replicate. This not only reduces the administration efforts for a consolidation model, but also reduces the storage requirements of such a model, as only one statement needs to be created for all of the nodes replicating to the central server.

In this case, the spoke servers are defined as send-only and replicate to the central server which is defined as a normal server. The spoke servers never replicate to one of the other spokes, nor does the central server replicate to the spokes. As a result, when **cdr check replicate** is run, there is no consideration of extra rows on the central server as an error condition, nor does differing sets of data between any two of the spoke servers generate an error. The only real error condition arises if there is a row on one of the spoke servers that does not exist (or is mismatched) on the non-spoke servers.

Monitoring Enterprise Replication Queues

Due to the asynchronous nature of Enterprise Replication, it is necessary to wait for a replication command to get executed at all Enterprise Replication servers before retrieving the command status. Unfortunately this wait period can be indeterminate.

A new Enterprise Replication queue monitoring feature, **cdr check queue**, has been added that will check queue status before attempting subsequent and dependent commands over the replication network. This provides a means of monitoring ER queues at one or more servers participating in replication and an option to monitor queues to be drained at all servers.

Control, send, and receive queues are the Enterprise Replication queues supported as part of this feature. They help take advantage of threading, to monitor multiple unrelated replication servers in parallel, and the relationship between servers. They also monitor the parent server before monitoring the child server and serialize queue monitoring for parent and child servers when hub

servers are involved.

The control queue, send queue, and receive queue can be monitored for being emptied as of the time of command execution, with the wait time being configured while monitoring Enterprise Replication queues to be emptied. The wait time (in minutes) can be defined and allows the Enterprise Replication queues to be synchronized up to a certain time, before executing subsequent commands over the Enterprise Replication network.

For the control and receive queues, the command waits for all messages in the queue up to the most recently queued message to be deleted from the queue. For the send queue, the command waits for all of the committed transactions that were active at the time the program was run to be queued into the send queue, and these transactions are deleted from the queue.

Data Replication for Servers with Different Owners

Elevated privileges are required to run most Enterprise Replication commands. Prior releases of Informix required connecting to database servers as user informix to initiate replication. Now it is possible to replicate data among database servers that have non-root installations and that do not have a user **informix** account. All that is required is to set the REMOTE_SERVER_CFG configuration parameter to a file that lists trusted hosts for remote authentication. In addition, all the servers in the replication domain must have the same owner.

To configure and manage Enterprise Replication, the administrator must have one of the following roles or privileges:

- Be the owner of a non-root server

- Have the Database Server Administrator (DBSA) privilege

- Be user informix (UNIX) or be a member of the Informix-Admin group (Windows).

Enterprise Replication Support for Light-Append Operations

Data can be loaded into or out of tables in a replication environment using a number of different methods. However, the faster mechanisms for loading use an unlogged, light-append operation which was not previously able to be replicated. With version 12.10, these unlogged changes to a table can be replicated to other tables.

If the replication environment has not yet been set, any of the standard load mechanisms can be used to load data: High-Performance Loader, **onunload** and **onload** utilities, **dbexport** and **dbimport** utilities, UNLOAD and LOAD statements, and external tables. To use load and unload tools on tables that are already being replicated, it is necessary to block replication while the table is prepared.

Integration with Automatic Storage Provisioning

The automatic storage provisioning feature, introduced in an earlier release of Informix, ensured there was significant reduction in the failures related to storage management and improved the self-healing capabilities of the server. However, failures related to storage management in an Enterprise Replication environment were still possible. If the user wanted Enterprise Replication to take advantage of the provisioning feature, the user had to manually configure the chunks used by Enterprise Replication databases and sbspaces.

The lack of integration between Enterprise Replication and storage management has been addressed in the latest release of Informix so that Enterprise Replication can make use of automatic storage provisioning. This new feature will automatically create and configure disk spaces as required by Enterprise Replication. If the server is also the primary of a high-availability cluster, then these same spaces and *onconfig* settings will also be appropriately configured on

all secondary servers.

Enterprise Replication requires a certain amount of disk space to be configured. The *onconfig* parameters CDR_QDATA_SBSPACE and CDR_DBSPACE were required to be set before the **cdr define server** command is executed. With this new feature, if the Informix server has a defined storage pool with sufficient space allocated to it, and even if the *onconfig* settings are unset, then executing **cdr define server** will perform the following:

- Create the required spaces from the storage pool
- Update the *onconfig* settings
- Proceed to define the new Enterprise Replication node.

The user need not create a dbspace or an sbspace while defining an Enterprise Replication node. The feature will dynamically add the spaces if the storage pool is full, and dynamically adds a chunk if the table space is full.

Improved Network Failover Support

The ability for a resilient system to recover from failure depends largely on the ability to detect that a failure has actually occurred. In a complex multi-node environment with multiple network transports, simply monitoring one network or testing for server availability through one interface might not be enough to detect a failure. It can appear to the Connection Manager (the monitor process that controls client-server connectivity and triggers failover) that the primary server is still up, but the primary server might not be available for clients that connect through a different network interface. Overall, it might be better to perform a failover and promote a secondary server that is available to those isolated clients. A number of new features in 12.10 provide a mechanism to achieve this greater level of recovery from failure scenarios.

Connection Manager Changes for Network Failover

In the event of network port failure against the primary database server, a local Connection Manager might still detect that the primary is active and avoid performing a failover. However, if the application is not able to get to the primary, then for all practical purposes, the system is still down.

Hence, placement of the Connection Manager plays an important role in identifying the issues with the network. To monitor the network connection between the Connection Manager and the database, the Connection Manager must be running on the same machine as the critical application servers. The Connection Manager is configured to monitor network communications between application servers and database servers. A server failover can be initiated only if the communication between the application server and the primary server is down. The Connection Manager will follow this sequence:

- Evaluate the impact of a failover based on all network monitor priorities

- Allow a failover if no higher priority network will lose cluster access

- Disallow a failover if any higher priority network will lose cluster access

- Allow an alternative means of communication for Shared Disk Secondary (SDS) clusters whenever the TCP/IP communication is disrupted or unavailable.

Alternate Communication for SDS Servers

A situation where the communication between an application and the database server is broken is an additional type of condition where a failover is required. In this case it would be desirable to perform a failover to a server that can be reached by the application.

This feature allows the primary and SDS nodes to communicate via an alternate communications path if network is temporarily unavailable. A new `onconfig` parameter, SDS_ALTERNATE, has been introduced which defines a dedicated blobspace for primary and SDS nodes to use for communicating. This allows the servers to continue functioning even when the network connection is down. The cluster can effectively be managed even without network communication between the nodes.

Server Side Support for Application Failover

The server side support for application failover feature ensures that if the application servers connected to the nodes in an SDS cluster are no longer able to connect to the primary node (even though the primary is still active), then an SDS server failover will be initiated and the SDS node will become the primary node and application servers will now have the connectivity to the new primary.

The support for application failover consists of having a Connection Manager running on the same machine as the application server. Each Connection Manager running on the application servers will periodically post network monitoring messages to all the database servers. The messages will contain information on the connectivity status from the application server to each database server along with a time stamp and the priority of the cluster network monitor.

It is now possible to manage the cluster even though there is no longer network communication between the nodes. The need to perform a failover arises only if the most critical application server no longer has access to the primary database. The priority of the application server needs to be understood as it would be unwise perform a failover if that means that a more critical application server would lose connectivity to the database in the process.

When a failover is requested, the failover node will access the network topology information to decide if failover is allowed to proceed or not. If a failover will result in a higher priority cluster network to lose access to the database cluster, then the failover is not allowed. Otherwise, a failover will proceed.

Embeddability and Administration

Ease of use has always been a primary goal for each new Informix release and a number of new features in 12.10 are specifically aimed at reducing administration overhead:

- Use of shell environment variables in configuration parameter values

- Simplified configuration files so only a few, critical parameters are required

- Turning automatic tuning on and off with one global configuration parameter

- Exporting and dynamically importing configuration parameter values

Simplified Server Configuration

Configure Automatic Tuning Parameters

A new configuration parameter called AUTO_TUNE has been introduced which will collectively enable or disable all the server features related to automatic tuning. Currently this includes AUTO_CKPTS, AUTO_LRU_TUNING, AUTO_AIOVPS, but the feature will encompass all automatically tunable parameters in the future. If the individual parameter is not present in the *onconfig* file, then it will take the AUTO_TUNE default. Otherwise, if present, their values will override that of AUTO_TUNE.

Modify Groups of Configuration Parameters

It is now possible to manage configuration parameters by exporting, importing, and modifying them in groups using **onmode**.

- **onmode -we**   To export a snapshot of the current configuration to a file for use as a configuration file that can be imported

- **onmode -wi**   To import parameters from a configuration file and modify those parameters in the file that are tunable

There are new SQL administration API commands to modify, reset, export, and import a configuration file. Use the SQL administration API command '**modify config'** to change the value globally, '**reset config'** and '**reset config all'** commands to revert the value of a configuration parameter to value in the last saved *onconfig* file and use modify '**import config'** or '**export config'** commands to import and export configuration files, respectively.

View the Current Values of Configuration Parameters

There are new **onstat** commands to view the current values of configuration parameters. Use **onstat -g cfg** to display configuration parameters and their current values, or **onstat -g cfg full** to display more information about each parameter, such as the default value, maximum and minimum values, and a description. The **onstat -g cfg diff** command will display only those parameters with current values that are not the same as the values in the *onconfig* file, and **onstat -g cfg tunable** displays only tunable parameters.

Setting Local Environment Variables for Informix Instances (UNIX)

A system administrator needs the correct environment variables set to be able to run Informix tools from the command line. For every Informix server, it was necessary to create an environment settings file, which needed to be sourced and set, before working with an Informix server instance. However, this practice makes running commands from remote machines difficult.

Informix 12.10 allows the setting of local environment variables in the `onconfig` file for an Informix instance. It is now possible to start the server and run Informix utilities without previously setting environment variables by reading those values from the `onconfig` file instead. All essential environment variable values such as INFORMIXSERVER and INFORMIXSQLHOSTS are read from an `onconfig` file specified in the command line as an additional argument to the **oninit** command when the server is started.

These settings are independent of the global or system environment variable settings on the computer. The settings can be used by the following utilities: **oncheck**, **onclean**, **oninit**, **onload**, **onlog**, **onmode**, **onparams**, **onspaces**, **onstat**, **ontape** and **onunload**. When running the Informix utility in this manner the new **-FILE=** option must be specified before any other options.

This feature will further improve the ease of use and embeddability of Informix as there is no longer a need to ensure all environment variables are established before executing the server start up command or worry about security issues that can be created by starting the server with an unknown environment. It is also possible to perform a variety of activities from remote servers, such as starting and stopping a remote database server and performing an **ontape** backup on a remote server, without having to create specific environment files and wrappers for each instance.

Managing Server Connections on Windows

The earlier releases of Informix on the Windows operating systems relied on the `HKLM\SOFTWARE\Informix\SQLHOSTS` registry key to store the connection information for Informix servers. In contrast, the same information is stored in the `sqlhosts` file on UNIX and Linux; on these operating systems, the `sqlhosts` path is commonly specified by the INFORMIXSQLHOSTS environment variable.

In Informix 12.10, Windows users configure connectivity information using the same file format as on all the other platforms. The file used for this is `%INFORMIXDIR%\etc\sqlhosts.%INFORMIXSERVER%`. The **syncsqlhosts** utility can be used to migrate connection information between an `sqlhosts` file and a Windows registry. A Windows registry can be used as either a source or a target, and it does not have to be the registry of the local machine on which the sync command is run. Windows allows for the updating of remote registries, assuming the user has appropriate permissions.

This feature provides two main benefits: system administrators no longer have to be familiar with different connectivity formats between different kinds of systems, and it is easier to distribute updates to connection information in typical, mixed-operating-system enterprises.

Enhanced Built-in Storage Management for Backup and Restore

Informix has two utilities, ON-Bar and **ontape**, that enable an administrator to backup and restore the server. The standalone **ontape** program is very simple to use but is a single-threaded process. ON-Bar does allow for parallelism but requires an external storage manager. Previously, a basic storage manager, Informix Storage Manager (ISM), was included with Informix, but ISM has an internal limit of 4 parallel threads, one per media device, and a 64 KB limit on the data transfer buffer size, which makes it unable to fully utilize the capabilities of modern media devices.

Informix 12.10 has introduced the Primary Storage Manager (PSM) to replace ISM. PSM is an application that manages storage devices used for backup and restore requests that are issued by ON-Bar. It is easy to use, with simplified maintenance and administration, allowing backups and restores to be performed at rapid rates, and is well suited for use in embedded backup and recovery solutions.

PSM performs data archive operations on regular dbspaces, blobspaces, sbspaces, and logical log files. The buffer transfer size no longer limited to 64 KB and PSM can write to all available media devices in parallel. The buffer size used by PSM is measured in memory pages, not bytes. By

default, 512-page buffers will be used, but buffers can be set to as large as large as 32,766 pages using the BAR_XFER_BUF_SIZE configuration parameter. On a 2 KB page system, the default buffer is 1 MB. PSM imposes no limit to the number of media devices that can be used.

Most of the capabilities previously supported with ISM continue to be supported with PSM. The **onsmsync** command can be used to support recycling and expiration policies. Encryption and compression can be managed through ON-Bar filters. The interface between ON-Bar and PSM continues to be X/Open Backup Services API (XBSA). One significant difference between ISM and PSM is that PSM does not support tape devices.

Functions of PSM:

- Create, modify, and delete storage devices
- Define and modify maximum sizes for devices
- Group devices into separate logical collections for log backups, space backups, and exporting or importing data
- Migrate backup information from one device to another within a pool
- Determine whether volumes, objects, and devices are locked or busy
- Release locked volumes, objects, and devices
- Verify volume names and labels
- Enable objects to be compressed and encrypted
- Expire and remove out of date backup information as directed by **onsmsync.**

Autonomic Storage Management for Rolling Window Tables

The interval table fragmentation strategy greatly simplified the storage management in an earlier release by allowing for automated creation of new fragments as needed. Informix 12.10 has new options enabling the ability to archive or destroy fragments based on configurable upper limits to either the number of fragments or the total table size.

The rolling window feature allows for the management of table space allocation without ever having the need to take explicit action. The ROLLING FRAGMENTS and LIMIT TO options of the interval fragment clause define upper limits on the allocated storage size or on the number of fragments in a rolling window table. Fragments that reach the limit are detached from the table, and can be either archived or destroyed. The automated management of storage space is useful in general, but can be especially beneficial when the Informix database server is used in an embedded solution where little or no database administration is a requirement.

Enhancements to the OpenAdmin Tool

The All New Welcome Page

OpenAdmin Tool (OAT) 3.11, installed by default with Informix 12.10, has an all new welcome page with links and resources to get started, customize the display and to help learn more about the tool. The welcome page also allows the user to configure the OAT home page.

Error! Reference source not found., shows the OAT welcome page section. Graphical displays of server performance and summary information for all the servers within a group can be chosen to be part of the OAT home page. Use the **Choose a custom home page** option to make any page within OAT be set as the home page. Navigate to the Dashboard through the Health Center → Dashboard page.
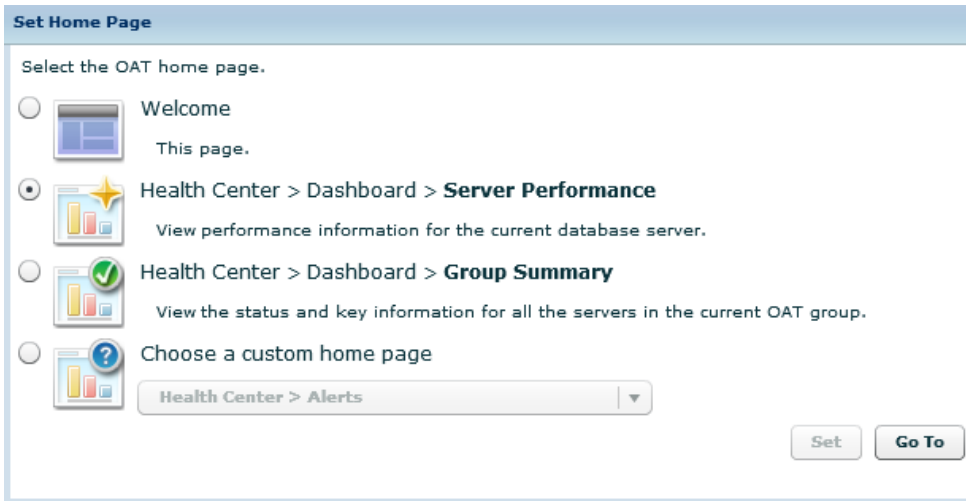
Figure 1: Set Home Page Options

Dashboard to Monitor Group Summary

Multiple Informix instances can be added to an OAT group and these servers can be monitored from a single location on the dashboard's Group Summary tab. This new multi-server dashboard monitors all the servers in a single OAT group simultaneously. The group dashboard displays summarized information on server instance status, including alerts, CPU, memory consumption, space usage, I/O activity, and backup status for all Informix instances associated with that group. **Error! Reference source not found.**, shows an example.

Server Name	Status	Alerts	Errors	CPU Usage	Memory	Space	I/O	Backups	Sessions	Last Upd...	Refresh
chicago	🟩	14	0	1.2%	🟩	🟩	🔶	🟩	3	15:29:10	🔄
amsterdam	🟩	15	0	1.81%	🟩	🟩	🟩	🟩	3	15:29:09	🔄
beijing	🟩	13	0	1.09%	🟩	🟩	🔴	🔴	3	15:29:32	🔄
cdrqa1	🟩	8	5	36.56%	🟩	🟩	🔴	🟩	5	15:29:33	🔄
hongkong	🟩	13	0	32.52%	🟩	🟩	🔴	🔴	4	15:29:34	🔄
newa	🟩	22	0	1.43%	🔴	🟩	🔶	🔴	3	15:29:33	🔄
server13	🔴	--	--	--	--	--	--	--	--	15:29:10	🔄

Figure 2: Group Summary Tab

The status information for each server instance is loaded asynchronously, which, along with the staggered refresh option, reduces delays in displaying the data. The dashboard uses thresholds to determine the alert status (red, yellow, or green) displayed in the various columns.

Dashboard to Monitor Server Performance

The real time status information is part of the Server Performance and Resource Usage dashboard. This dashboard consolidates information such as CPU and memory usage, space allocation, the current number of locks and transactions, and the tables with the largest amount of activity from the given database server instance. It allows users to monitor the status of many remote servers by using one dashboard from a centralized location.

The dashboard is also customizable. It can be changed using the panel buttons on the page or by adding dashboards on the fly. It is possible to create a new dashboard and then customize it by choosing all, or a subset, of the panels from the available set.

Support for Users Other Than "informix"

OAT v3.11 has improved security and access control features for database users. Additional administrative users (other than user **informix**) can be defined rather than having several users sharing a single Informix administrator account to login to the system.

It is possible to grant and revoke administrative privileges to separate users, providing finer granularity on authorizing individual users to perform different tasks. Users with these privileges can connect to a database server with their user name and run SQL administration API commands, or perform those tasks using OAT.

Use the Server Administration → User Privileges → SQL Admin API Privileges page to manage the individual user privileges on the database server.

Generating Heat Maps

OAT v3.11 shows all the server information with an easy-to-read graphical format display. It is possible to choose to view the graphs of buffer pools and extents of a given database server or the percentage of cached pages in the buffer pools for databases. This type of graph is called a heat map or a tree map. The size of a block represents the number of elements within that set relative to the other sets, and "hotter" colors are used where the percentage of items in use within a set is higher.

Use the Performance Analysis → System Reports → Graphs page in order to generate a heat map. The drop-down controls at the top of the page indicate the type of data to graph (extents or buffer pool) and particular database to analyze. The size and colors on the map represent the extents and buffer pool usage on the selected database servers.

Extent graph – Each rectangle within a displayed database represents an object, such as a table or an index. The size of the rectangle indicates the relative size of the object. The color indicates the number of extents as indicated by the legend for the graph. The number of extents can affect the ability of the database server to scan a table rapidly for data. The greater the number of extents, the more fragmented the data is across a device and the longer the time spent in seeking the data.

Buffer Pool graph – As with the extent graph, the buffer pool graph also presents each database object, with the size of the rectangle indicating the relative size of the object. The color indicates the percentage of pages that are cached as indicated by the legend for the graph. A higher percentage of cached pages can improve performance. Queries run on tables with a high cache percentage are faster than queries run on tables with a low cache percentage.

Compression has New Dimensions

Informix 12.10 has enhanced its compression technology by extending it to compress B-tree indexes and simple large objects when those large objects are stored in the same space as the other table data. Compression can now be enabled automatically as part of a table's definition; however, there are lower bounds on the number of rows, below which there is little to be gained by compression, and the amount of data must be greater than that before the table is actually compressed.

OpenAdmin Tool – Compress a Simple Large Object

On the Space Administration → Storage → Tables and Indexes page it is also possible to compress simple large objects as well as compressing tables and fragments. It is also possible to estimate the amount of space savings to be achieved before actually performing the operation.

OpenAdmin Tool – Compress B-Tree Indexes

Compression of indexes can be performed from the same page as compressing tables. It is also possible to compress an index at creation time by selecting the compressed option on the Create Index Wizard in the Schema Manager plug-in. An estimate on the amount of space that can be saved when compressing an index can also be generated.

Along with the compression activity, it is possible to choose to consolidate free space by repacking the partition and then returning the free space consolidated at the end of the dbspace by choosing the shrink option.

Access the Index compression functionality, as well as the compression estimate tool, by navigating through OAT → Space Administration → Storage.

Automatic Backup

Earlier releases of OAT supported the scheduling of backups using only the **ontape** utility with the option to specify a tape device, a file on disk, or a directory as storage mechanisms. The ON-Bar utility wasn't supported for scheduling backups.

With the introduction of Primary Storage Manager (PSM) as the new built-in storage manager, replacing the Informix Storage Manager (ISM), OAT now includes a Backup Wizard to configure automatic backups. The wizard uses four simple steps, to choose the backup utility (ON-Bar or **ontape)**, set basic and advanced configuration parameters, and confirm the backup scheduling. When using ON-Bar it is necessary to choose a storage manager such PSM or another XBSA compliant storage manager.

Enhancements to the Schema Manager

Manage Regions and Grids along with Grid Tables

OAT has been enhanced to work with the new Informix 12.10 features, grid enabled queries and grid tables. Grid tables can be quickly defined by graphically adding or removing tables to or from the grid table list for a selected grid. Grid tables must have the same column names and attributes on all grid servers and can be used in queries across grid tables. These queries run on each server in a region or a grid and the results are consolidated together [Refer to the topic Simplified set up of a Consolidation System in the Cloud section above for a detailed description].

To view and define grid regions using OAT, navigate to the Replication → Grid page. To add or remove tables to or from a grid table list, select options from the Action menu on the SQL Toolbox → Schema Manager page. Run queries across a grid or region from the SQL ToolBox → Query By Example page.

Manage Fragmentation

OAT has been enhanced to support the new Informix 12.10 feature, rolling window tables. When defining tables with interval table fragmentation it is possible to configure the policy and identify and mark the fragments eligible for removal with rolling windows. Choose also whether only interval fragments, or both interval and range fragments are removed, and optionally to have range fragments removed only after all the interval fragments are removed.

Configure which fragments are eligible for removal when creating a table on the SQL ToolBox → Schema Manager page, and then select a date-range fragmentation or range fragmentation storage scheme.

Administering Informix Warehouse Accelerator using OAT

Informix Warehouse – Welcome page in OAT

Navigating to the Server Administration → Warehouse Accelerator page in OAT leads to a Welcome page for the Informix Warehouse Accelerator. This welcome page is a useful summary of the resources allocated to the accelerator.

Manage Accelerators

An accelerator server is an installation of Informix Warehouse Accelerator and each accelerator contains information on connections between the accelerator server, the database server, and the data marts associated with that connection. Each accelerator server can have one or more accelerators.

With the Informix Warehouse feature in OAT, there is the ability to manage and monitor accelerators by including tasks like verifying the configuration details of the machine and accelerator instances, adding or dropping an accelerator, viewing the status of an accelerator, and viewing the data marts.

To manage and monitor the accelerators, navigate to the Server Administration → Warehouse Accelerator page in OAT. The warehouse page lists the accelerator server hosts and the accelerators that belong to them in the left navigation tree.

Manage Data Marts

Management options for Informix Warehouse Accelerator allow the ability to connect an Informix server instance to multiple accelerator servers. It is possible to create any number of data marts that are subsets of a database. While a database can have multiple data marts, a data mart can be deployed on only one accelerator. However, data marts from a given database can be deployed across multiple accelerator servers.

It is possible to create and load data marts and view the data marts associated with a database on the SQL ToolBox → Schema Manager page. The available operations are to create a data mart, schedule full loads, schedule loads to update only the changed partitions, view the mart information (such as status, schedule info and tables that constitute the mart), and enable or disable marts.

It is also possible to create a data mart through query workload analysis. To do this, a query workload is run on the server, which is recorded using the SQL tracing option, and is then analyzed to create the mart in a chosen accelerator.

IBM Mobile OpenAdmin Tool for Informix

The Informix 12.10 release launches the IBM Mobile OpenAdmin Tool for Informix, an app available for free download on iOS and Android. Mobile OAT is a display and monitoring option and not an alternative to the full, web-based OpenAdmin Tool. Users would still need direct server access to invoke Informix utility programs or OAT to act on information provided by Mobile OAT.

Use the Settings option on the OAT Mobile app to configure individual server instances as well the server groups, and customize them as required.
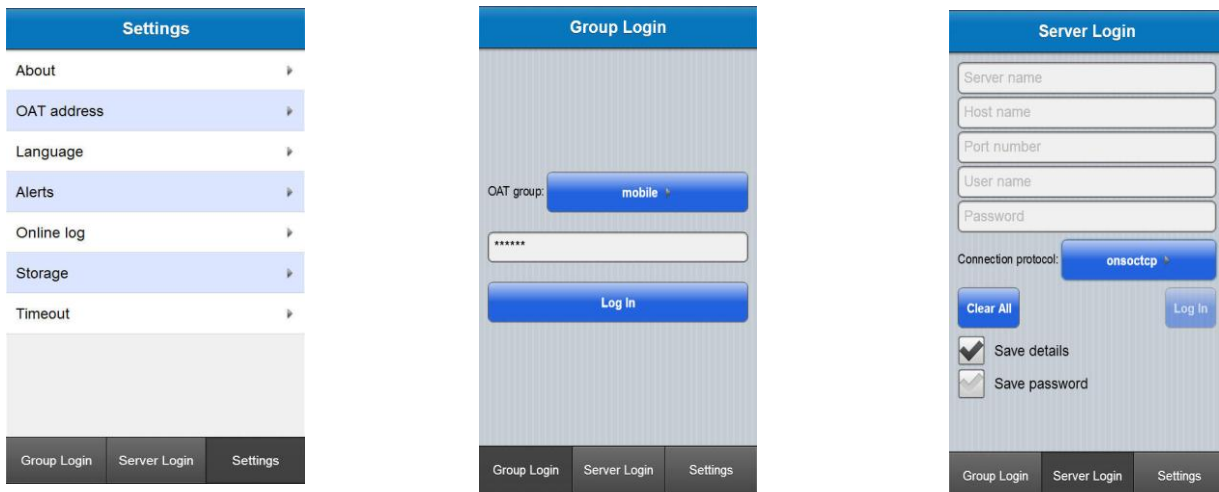
- **Settings –** Provide the OAT server address, which will be the web based URL to access OAT. The app can then be customized to set alerts, monitor the log, storage and define timeout periods.

- **Group Login –** Configure group access and use the group login credentials to access multiple server instances listed under the defined group.

- **Server Login –** Login to a specific Informix instance using the server login details and appropriate access levels.

The initial login page is the Health page of Mobile OAT app, with Users, Details, and Performance being the other three tabs available.
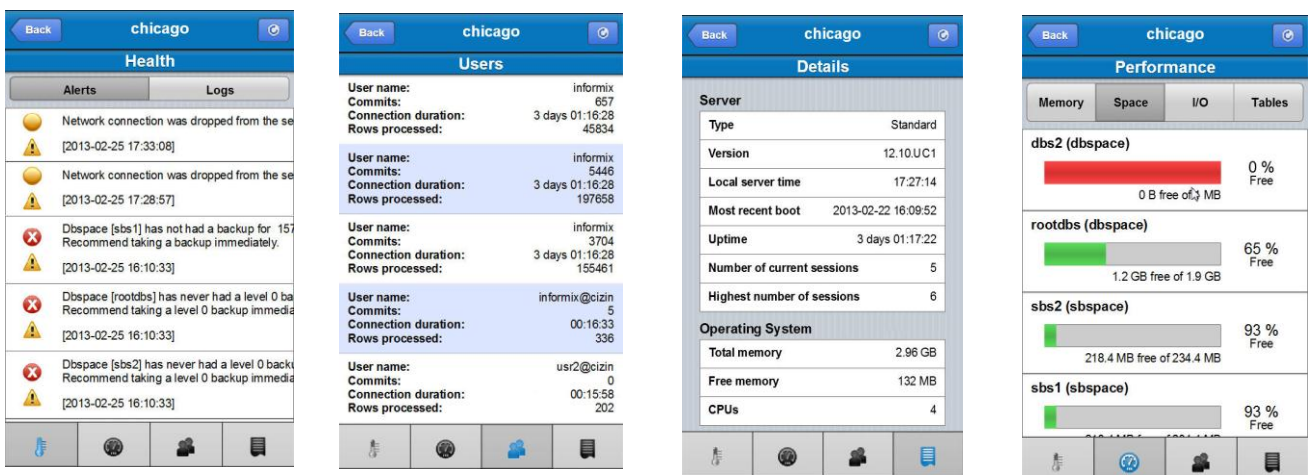
- **Health** – Lists alerts and logs.

- **Users** – High level information on the user and the activities carried out by them when logged in to the Informix server.

- **Details** – Summarizes the Informix server information including some information about the machine on which the server is running.

- **Performance** – Captures various aspects of the Informix server that helps in gauging performance, such as Memory, Space, I/O, and Tables.

Application Development Support

Language Enhancements

SQL Statement Length

Previously the length of SQL statements sent between client and server was limited to a maximum of 64 KB. Informix version 12.10 increases this limit to 2 GB.

This increase in size is important for creating Stored Procedure Language (SPL) routines that exceed 64 KB in length and for handling long and complex SQL statements. While it would be unusual for a developer writing SPL or SQL statements to exceed the previous limit, it is increasingly common for code generators within BI clients to build very long SQL statements. Removing this limit allows greater compatibility with the range of BI and other developer tools that are available.

Store Query Results Using the SELECT INTO Statement

Informix has been supporting creation and population of external tables and temporary tables using the projection list of a SELECT clause, but there was no such support available for permanent tables.

Informix 12.10 bridges the gap in migrating either from Extended Parallel Server (XPS) or other database systems to Informix by enhancing the SELECT INTO statement. It combines the functions of CREATE and INSERT into a single statement, enabling the loading of data retrieved by the SELECT statement into a new table. With this SQL enhancement, it is now possible to:

- Create persistent tables that can be accessed across sessions using an SQL result set
- Create a new schema from existing tables and populate it with only a subset of data for testing purposes
- Support all the storage options available with CREATE TABLE
- Use it in the Flexible Grid environment.

Using the CASE statement in SPL

Informix 12.10 allows SPL routines to use the CASE statement as a simpler alternative to multiple IF statements to define a set of conditional logical branches. CASE statements were previously supported only in SQL statement syntax, not in Stored Procedure Language (SPL). The new syntax simplifies migration of SPL in applications that were written for XPS or other database systems.

The CASE statement can be used in a stored procedure to take a different branch of execution based on the value of an expression or the value of a SPL variable. This will reduce the work required to migrate stored procedures from other databases and simplify programming by replacing multiple IF-ELSE IF statement blocks with a single CASE statement.

INTERSECT and MINUS Set Operators

UNION, INTERSECT, and MINUS are set operations that combine the results of SQL SELECT statements. The traditional UNION and UNION ALL operations were supported in previous Informix releases to get combined results of two or more queries, where the ALL keyword will allow duplicate rows to be included in the output.

Informix 12.10 now supports the INTERSECT and MINUS set operators in queries.

- INTERSECT operator – Computes the set intersection of the rows returned by the SELECT statements involved. The final result set has rows that are in both sets.

- MINUS operator – Computes the set difference of the rows returned by the SELECT statements involved. The final result set has rows that are present in the first set that are not present in the second set.

In both cases the result set will always be distinct, with duplicate rows being eliminated.

Multiple DISTINCT Aggregate Functions in a Query

Informix previously allowed only a single DISTINCT aggregate in SQL queries, forcing users to write multiple single distinct aggregate queries across temporary tables to achieve the required aggregation result.

Informix 12.10 eliminates this restriction by allowing multiple distinct aggregates to be specified in SQL queries. The benefits of this new aggregation feature include:

- All the existing aggregates and support for built-in data types allowed with single distinct aggregates are supported for multiple distinct aggregates.

- Result sets that used to require the writing of multiple statements can now be written with one.

- It is supported with PDQ (Parallel Data Query).

- Migration efforts from XPS and other database systems to IBM Informix are easier.

- The ability to express in one statement the same result that otherwise would take more than one yields performance benefits.

Controlling the Sorting of NULL Values

Previous Informix releases always sorted NULL values last in the result set of an ORDER BY operation, no matter what the specified sort order was. Informix 12.10 now supports the ISO-9075 SQL Standard for specifying the placement of NULL values in a sorted result set. New keywords NULLS FIRST or NULLS LAST can be used to specify the result set order of rows that have a NULL sort key value. As well as being more standards compliant, it is now easier to generate results that are compatible with other database servers.

SQL Package Extension

The SQL package extension provides routines that can be used in an application intended to be compatible with database servers other than Informix. The following modules are included:

- DBMS_ALERT – Provides a set of procedures for registering for alerts, sending alerts, and receiving alerts

- DBMS_LOB – Provides additional operations on large objects

- DBMS_OUTPUT – Provides a set of procedures for putting text in a message buffer and getting messages from the message buffer

- DBMS_RANDOM – Provides a mechanism for generating random numbers

- UTL_FILE – Provides a set of routines for reading and writing to files on the database server file system.

Enhancements to Genero

While not bundled with Informix 12.10, IBM Informix Genero is a key integrated development environment intended for developing applications to work with Informix 12.10. Genero is on its own release schedule, with Genero 2.41 being the most recent release. The Genero 2.41 release has made improvements to the HTML5 theme for Genero Web Client (GWC) and has added new templates for the Business Application Modeling (BAM) tool.

Enhanced HTML5 Capabilities for Genero Web Client

The HTML5 theme is the AJAX theme rewritten to take advantage of HTML5 and CSS3 improvements. Intended to work with the latest web browser releases and mobile devices such as tablets and smart phones, this theme gives a richer user experience. Genero Web Client (GWC) support for HTML5 uses native widgets, according to the device on which the client application is displayed. It is recommended to use the HTML5 theme for any new application development.

The HTML5 theme highlights the improvements done to the screen layout, support for Genero presentation styles (.4st files) and TTY attributes, methods to facilitate migration to HTML5 from the AJAX and Silverlight themes, and more.

The Genero Hybrid Client

The Genero Hybrid Client (GHC) provides the ability to launch an application within a container native to iOS or Android. This allows Genero HTML5 web applications to be launched and rendered so that they look and act more like native applications for Android or iOS mobile platforms. GWC also provides templates, which when customized, can interact with services on the local device. This allows the Genero application to interact with the device's location based services or access local storage, for example.

New Application Generation Templates

Informix Genero Studio provides standard template sets containing files used for application generation. A new default database application template, dbapp2.0, includes much more powerful build and generation rules that correlate a customized set of build rules to a template. Written in the TCL language, these Business Application Modeling templates can be modified to meet specific business application requirements.

Performance Enhancements

Increased Scalability with Optimized Caching

Informix uses a number of caches to speed up access to various types of internal data structures. Once a cache entry element is created, the server simply accesses it without incurring the cost of creating it again. The SQL cache and data dictionary caches are known to be volatile caches; therefore, optimizing the cache access methods and cache maintenance can improve performance.

Informix 12.10 increases the scalability on systems with many sessions or virtual processors by introducing the following:

- R/W locks – These allow multiple, simultaneous readers, which reduces contention and maintenance of cache entries; they are used to replace mutually exclusive (mutex) locks where possible.

- Atomic counters – Counters can be maintained with atomic instructions that have less contention than using a latch.

- Smarter heuristics – To help maintain the cache usage pointers and reduce cache purging. This reduces the frequency of exclusive access requests and so increases the opportunity for shared access to improve performance.

These internal enhancements have no externally visible impact, other than increased performance.

Faster ANSI Join Queries

Informix 12.10 can now use dynamic hash joins for ANSI outer join queries. Previously with ANSI format queries, the server would only consider filtering and join options based on predicates specified in the WHERE clause. In such cases a nested loop join was the only method available to perform ANSI joins.

Now, by utilizing information in the ON clause, the optimizer can consider other join methods. ANSI join queries with equality joins in the ON clause can be candidates for either a dynamic hash join or a nested loop join, based on their relative cost. The support for ANSI joins is implemented against left deep join trees and against mixed (bushy) types of left and right deep trees. A hash join will be tried wherever left deep join trees are found.

When the ON clause filters are not satisfied, the NULLs are extended to the subservient table and then the filters in the WHERE clause are applied on this result set. A dynamic hash join will not be considered when the ON clause does not contain an equality join between the tables.

Temporary Table Projection Optimization

Applications and analytic tools can define queries requiring intermediate results that are materialized by the database server in temporary tables. For complex queries, the derived tables could contain multiple views joined with regular tables, potentially having hundreds of columns, although the parent query might project only a few columns. This often means that the additional columns will take up a lot of unnecessary temporary space in the server, and the optimizer may overestimate the size of the temp tables for derived views and miscalculate the FIRST extent size, resulting in a disk space shortage.

Informix 12.10 generates temporary tables internally that include only the columns that are specified in the final projection list, the WHERE clause, the ORDER BY clause, and in other clauses of the immediate parent query. By excluding the unnecessary columns from the temporary table, the database server uses storage resources efficiently and avoids I/O operations on the columns that do not contribute to the query result.

Extended Platform Support

Support XML Functionality on Windows 64-bit platform

Support for XML is an existing feature on Linux, UNIX and Windows 32-bit platforms. However, this support (XML Parser, support for XPath, XSL support and others) was not extended to the Windows 64-bit platform.

Informix 12.10 now extends the support for XML, making it available on the Windows 64-bit platform. The feature has also been back-ported and is made available in the Informix 11.70.xC6 fix pack.

Support for Unicode 6.0

Informix 12.10 is now supports Unicode 6.0, the latest version of International Components for Unicode (ICU) for globalization and Unicode. This upgrade covers both the server and CSDK and takes the latest fixes from ICU and implements new features in compliance with the latest

specification of Unicode.

References

This document was prepared in conjunction with the following IBM Informix publications:

Administrator's Guide

Administrator's Reference

Guide to SQL: Syntax

IBM OpenAdmin Tool (OAT) for Informix

Embedding IBM Informix

TimeSeries Data User's Guide

Spatial Data User's Guide

Backup and Restore Guide

Informix Warehouse Accelerator

Performance Guide

For more information

To learn more about the Informix features, contact your IBM representative or IBM Business Partner, or visit the IBM Informix web site.

Appendix – Informix Edition Changes

Informix comes in various editions which provide different levels of scalability and feature content. Some of the editions that are available at version 12.10 have different names and features since the 11.70 release. The following tables indicate where name changes have been made and the current availability of selected features.

Version 11.70 Edition Name	Version 12.10 Edition Name
Informix Choice	**No longer available**
Informix Developer	Informix Developer
Informix Innovator-C	Informix Innovator-C
Informix Express	Informix Express
Informix Growth	**Informix Workgroup**
Informix Growth Warehouse	**Informix Advanced Workgroup**
Informix Ultimate	**Informix Enterprise**
Informix Ultimate Hypervisor	**Informix Enterprise Hypervisor**
Informix Ultimate Warehouse	**Informix Advanced Enterprise**

Table 1: Edition Name Changes. Names in bold have been changed from the previous release.

The majority of all features, including the OAT, are available across all editions; the availability of selected features is identified in the following table.

Features	Developer	Innovator-C	Express	Workgroup	Advanced Workgroup	Enterprise	Enterprise Hypervisor	Advanced Enterprise
Primary Storage Manager	✓	✓	✓	✓	✓	✓	✓	✓
Enterprise Replication	✓		✓	✓	✓	✓	✓	✓
High Availability Clusters	✓		✓	✓	✓	✓	✓	✓
Shared Disk Secondary	✓		✓	✓	✓	✓	✓	✓
Remote Stand-alone	✓		✓	✓	✓	✓	✓	✓
Updatable Secondary	✓	✓	✓	✓	✓	✓	✓	✓
Parallel Operations	✓					✓	✓	✓
Data Partitioning	✓					✓	✓	✓
Private Memory Cache VP	✓					✓	✓	✓
High Performance Loader	✓					✓	✓	✓
Distributed Operations	✓					✓	✓	✓
Compression	✓					Optional	✓	✓
Informix Warehouse Accelerator					✓			✓

Table 2: Selected Feature Availability by Edition

IBM Informix 12.10 *New Features* **March 2013** Cosmo David Chris Golledge Prasanna A Mathada © Copyright 2013 IBM Corporation IBM Corporation Software Group Route 100 Somers, NY 10589 U.S.A. The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In	IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results. IBM, the IBM logo, ibm.com, and Informix are

addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice.

trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.