Why upgrade to IDS 10.0 – A Technical Perspective

by Jacques Roy

This document discusses the reasons why someone using an older version of IBM Informix Dynamic Server (IDS) would want to upgrade to IDS 10.0. As stated below, IDS 9.21 is a superset of functionality from IDS 7.31. By considering features included in the IDS 9.x code line in different versions, we can find the new features that were added after the IDS 7.x code line. By dividing the features by release, it makes it easier for anyone to have a better idea of how much more they get by going to IDS 10.0. It should be noted that the upgrade to IDS 10.0 is a direct step from IDS 7.24, 7.30, 7.31, 9.21, 9.30, and 9.40.

The merging of functionality

The IDS 9.x code line started from IDS 7.12. Both code lines (7.x and 9.x) continued to evolve separately until IDS 9.21. Starting with IDS 9.21, released in March 2000, the IDS 9.x code line became a superset of IDS 7.31.

IDS 7.31 came out in February 1999. Since then it has gone through 15 releases, most of them being bug fix releases. Since the merger of functionality (7.31UC6 and 9.21), the new functionality included in IDS 7.x consist of back-porting features from newer releases of IDS 9.x. This includes features such as SQL optimization, performance optimization and the new B-tree scanner. The B-tree scanner was also back-ported to IDS 9.30UC7 from IDS 9.40xC1. The current release of IDS 7.31 is 7.31.UD10. It became available in August 2006.

> *IDS 10.0 is a superset of functionality of any previous version of IDS. It is the logical upgrade path for all previously supported versions of IDS.*

Ease of Upgrade

As mentioned above, upgrading to IDS 10.0 is a direct upgrade from multiple versions of IDS. The upgrade can be quite simple. After installing IDS 10.0, one only has to bring up the server and the upgrade is completed in a few minutes during the server initialization. The only concern should be the new keywords added between the current version used and IDS 10.0.

In addition to the ease of upgrade, an IDS customer stated that his 2.5 million lines of Informix 4GL did not even need to be recompiled to work with IDS 10.0.

For more details on upgrading to IDS 10.0, please refer to: IBM Informix Migration Guide. This guide can be found on the documentation site listed in the reference section.

Criteria for upgrade

The reasons to upgrade IDS include performance and scalability, reliability and availability, manageability, and application development features. These criteria are reviewed in detail later in this document.

One criterion that is not listed is database extensibility. Database extensibility was the reason for the creation of the IDS 9.x product. IDS extensibility allows for a better integration of the database server with the business environment. This results in simpler solutions and faster performance. IDS extensibility has been used to solve business problems in airline capacity planning, retail, and government agencies among other areas. The choice of IDS as the underlying database for the WebSphere RFID Information Center product was made in great part because of its extensibility features.

We see extensibility in all sorts of products including operating systems, web servers, and integrated development environments (IDEs) such as Visual Studio and Eclipse. We could even consider application servers as a formalized extensible platform. A recent Infoworld[1] article on a successful SOA project lists, in its performance recommendation, that data intensive processing should be put in the database. This makes the case for greater use of database extensibility.

IDS extensibility is outside the scope of this document. You could consider extensibility, at a minimum, as the ability to write stored procedures in "C" or Java. Please take some time to look at the examples provided in the reference section.

Performance and Scalability

Performance is often the number one concern of many IDS users. It took until IDS 9.40 to put this issue completely to rest when comparing IDS 7.x to IDS 9.x. Despite some performance work on IDS 9.x, some operations remained faster in 7.31 in releases before IDS 9.40. One of the major goals of IDS 9.40 was therefore to be faster than IDS 7.31 in all situations. This goal was achieved by doing nightly performance test comparisons during the development cycle. Any performance issues found during the nightly tests were aggressively pursued. This resulted in a performance improvement that varied between 15% and 20%. This effort continued in the development of IDS 10.0 where we see IDS 10.0 being faster than 9.40. These performance tests are now part of the development process where each new release of IDS must be faster than its predecessor.

Internal tests have shown IDS 9.40 to be 15% to 20% faster than version 7.31 and IDS 10.0 is 5% to 10% faster than IDS 9.40

Going hand-in-hand with performance is the ability to scale as the load increases: scaling with more resources and scaling in terms of the volume of the data managed. The following table summarizes some of the major performance and scalability features that were added to IDS over time.

IDS 9.21 (and below)	IDS 9.4
- Lightweight I/O for BLOBs - Fuzzy checkpoint	- Decimal values for LRU min and max

---

[1] Infoworld, August 14, 2006
[2] back ported to IDS 7.31.UD8 and 9.30.UC7

- SQL statement cache	- More ER performance
IDS 9.30	improvements
- Improved space allocation of user	- DS_NONPDQ_QUERY_MEM
data and metadata in sbspaces	**IDS 10.0**
- ER Performance improvements	- Configurable page size
IDS 9.4	- Shared memory > 4GB
- B-tree scanner[2]	- DS_QUERY_MEMORY[3]
- Chunks up to 4TB	- Multiple fragments per dbspaces
- Number of chunks (32767)	- Indexes on non-logging tables
- Increased in server aliases	- Faster recovery from fuzzy
- Increased file-size limits	checkpoints
- PDQ enabled on hold cursors	- Improved performance on backup
- Improved buffer management	and restore
	- Enabling view folding

Without going into all the details of all the features listed above, a further explanation of some of these features better shows their importance:

SQL statement cache: The SQL statement cache allows for the sharing of SQL statements between user sessions. In the appropriate environment, it can reduce the time spent in parsing and optimizing statements and also save server memory.

ER Performance: Enterprise replication performance work provided huge returns in IDS 9.30. Some tests have shown a performance improvement from 300 Transactions per seconds (TPS) to 1200 TPS. The performance work continued in later releases.

Chunks improvements: the chunk size and maximum number of chunks removed the 4TB limit of addressability of an IDS instance and basically put it to a limit that is, today, practically unlimited. IDS can now theoretically address 128PB of data (128,000TB). The current physical limit is 32TB. The new chunk size can help simplify system administration for large systems.

Buffer management: This is an important feature that greatly reduced the complexity of our buffer management and improved its performance.

Configurable page size: This provides I/O performance benefits for the manipulation of tables that have large rows.

DS_QUERY_MEMORY: This parameter, which can be set dynamically, can change the default amount of memory allocated to an OLTP query. This allows the resolution of queries in memory instead of having to use temporary space on disk. This results in higher performance.

---

[3] Rename of the 9.40.UC4 parameter DS_NONPDQ_QUERY_MEM

Multiple fragments per dbspaces: This feature extends IDS's partitioning capabilities. By partitioning tables using a finer grained partitioning, you can improve fragment elimination. The result is less disk I/O, increasing performance.

The IDS multi-threaded architecture provides the foundation to exploit the capabilities of hardware platforms and operating systems. This results in excellent performance and scalability. The additional features added over time in IDS 9.x and 10.0 improve significantly the performance and scalability of IDS. Depending on the processing environment, the features listed above could have significant impacts.

Benefits of Performance and Scalability

Internal tests have shown that IDS 9.40 is 15% to 20% faster than IDS 7.x and that IDS 10.0 is 5% to 10% faster than IDS 9.40. Another way to look at it is that on an 8-CPU system, having a 12.5% performance increase is the equivalent of getting one additional CPU of processing power. At a 20% performance improvement, it is the equivalent of getting one additional CPU for each five CPUs. We could also look at it in terms of additional throughput, additional users and so on.

> *"We are now handling about double what we were able to handle under the old platform. So we have multiple servers that are handling a portion of these databases. Searches that were taking five minutes, 10 minutes before are now taking seconds."*
> Eric Gordon, Managing Director RealPlus LLC,
> on upgrading from IDS 7 on SCO to IDS 10 on Linux

This performance improvement can delay costly hardware upgrades and additional software licenses required with it (additional CPU licenses). A hardware upgrade can be disruptive. Being able to delay an upgrade or, in some cases, avoid it all together reduces the disruption to the business.

Performance could be further enhanced by taking advantage of additional features such as the performance features mentioned above, SQL enhancements and user-defined functions. New approaches to problem solving could translate in significant additional performance benefits.

With the proven performance boost and potential for more performance benefits, upgrading to IDS 10.0 makes a lot of sense.

Reliability and Availability

Code quality has a significant impact on reliability and availability. IDS 7.31 is already known as a very stable product but the IDS development team wanted to do more. The IDS team created new development processes to insure that IDS 9.4 became the highest quality IDS product at the time of its release. IDS 10.0 benefited from these new processes. It is now the best IDS release to date.

The development team benefited from additional hardware resources. This allowed them to run QA and performance tests nightly on all the supported platforms. With this capability, they instituted a process where new code, checked into the IDS code line, is tested the following night. If the result of the testing shows any regression or performance problems, the responsible engineer has four hours to solve the problems or he/she has to remove the changes and resolve the problem in his/her development environment before trying to integrate it again in the main code line. The benefit of this approach is that the release cycle does not depend on a large QA exercise when it gets close to the release date. Code quality is an ongoing concern and a priority.

The reliability of IDS is shown through examples such as a top retailer who happens to have over 10,000 instances of IDS world wide, achieving 99.999% up time (including planned downtime), or customers with instances up for over 1000 days and counting.

The latest releases of IDS also add features that improve reliability and availability. Some of major features are listed in the following table:

9.21 (and below)	10.0
- Dynamic lock allocation 9.3 - Dynamic log allocation 9.4 - Encrypting communication - ER security - ER large transaction support - ER dynamic log file - HDR and ER concurrently - Dynamically monitor queries (onmode -Y) - No files installed in /usr/lib - External authentication with Pluggable Authentication Modules - HPL improvements - Restartable fast recovery	- Column-level encryption - Server utilities check for secure environment before starting - Restricted registration of external routines - Preventing denial-of-service attacks - Replicating indexes to secondary server - Automating HDR failover - ER improvements - Creating and Dropping Indexes without Locking Tables - Recovering Quickly with Fuzzy Checkpoints - DBCREATE_PERMISSION parameter - SECURITY_LOCALCONNECTION configuration parameter - DB_LIBRARY_PATH parameter - Truncate Table

This table includes security features because compromised security can result in an unavailable system. For example, addressing denial-of-service attacks can be considered both a security and an availability feature.

IDS 10.0 is the most reliable IDS release ever!

The high-availability features of IDS include the ability to dynamically modify configuration parameters, the attach/detach feature for fragmented tables, and many more. It also depends on the capabilities of High-availability Disaster Recovery (HDR) and Enterprise Replication (ER). Both HDR and ER have been improved over the last few release in terms of capabilities, ease of manageability and performance. IDS 9.40 also adds the capability to use both products concurrently. This opens the door to new levels of high-availability as demonstrated by customers such as TCH (see the success story listed in the references section).

We could also add the new buffer management introduced in IDS 9.40 as a reliability and availability feature since it allows IDS to better adjust to the needs of the running system. This feature removes the need for the "SET RESIDENCY" SQL statements since IDS can now adjust dynamically based on the real-time needs of a running system. The new B-tree scanner is also much more reliable than the old B-tree cleaner used before IDS 9.40. We could make the case for a lot of the performance features as also being reliability and availability features.

IDS 10.0 improves on an already great story about availability and reliability. There is more to it than a list of features. Customers of any size can benefits from IDS 10.0 in this area.

Benefits of Reliability and Availability
The new IDS features provide the ability to insure that a mission-critical system is available even in cases of disasters through the use of High-Availability Disaster Recovery (HDR) and Enterprise Replication(ER).

Many features add autonomic capabilities to IDS. This way, the server can adapt to the needs of the running system and avoid things such as long transactions that might force a restart of the database server. This way, the database server is better able to respond to the variations of the work load and provide continuous services to the business.

It is important to note that many businesses lose money if their system is not available. Just think of a company that performs 100,000 transactions per day. If we give an average of $25 per transaction, we can say that one hour downtime would cost approximately $10,400. This is when we spread the 100,000 transactions over 24 hours. The cost is much higher when we consider peak times. IDS' improved reliability and availability directly affects the business' bottom line.

> *"Thanks to IDS, we relocated our entire data center with less than seven minutes of database downtime. We would never have been able to do that with Oracle."*
> Stephen Cobb, Senior Administrator of Systems and Databases, TCH

This section also included security features. It is worth mentioning that features such as column-level encryption and the ability to use external authentication with Pluggable Authentication Modules (PAM) can provide the needed functionality to address regulatory requirements.

Manageability

IDS is renowned for its ease of administration. We see this through customers that use a handful of database administrators to manage hundreds if not thousands of IDS instances. Manageability is one major reason why IDS is IBM's database of choice for industrial strength embedded OLTP systems.

IDS 10.0 includes new features that give a better view of what the server is doing. It also includes new capabilities to impact the optimizer such as the external optimizer directives.

The manageability benefits of IDS make it a great candidate for SMB enterprises. An IDS database can become virtually invisible. Its management can be done through scripting and some functionality makes it adapt to the running environment. The following table shows some of IDS's major manageability features:

9.21 (and below)	10.0

- Dynamic lock allocation
9.30
   - Dynamic log allocation
   - Display query plan without execution
   - IFX_DEF_TABLE_LOCKMOD environment variable
9.40
   - Configurable event alarms
   - Full use of storage media
   - Chunk reserve pages in non-root chunks
   - Restartable fast recovery
   - Display parameters of prepared statements
   - Rename chunks during restore
	- Configurable page size
	- Defining buffer pools
	- Managing tblspace tblspace
	- Single user mode
	- Managing database permission through default roles
	- Renaming dbspaces
	- Specifying which events trigger the alarm program
	- External optimizer directives
	- Storing Multiple Table or Index Fragments in a single Dbspace
	- ER master replicates
	- ER replicate templates
	- ER Altering Replicated Tables
	- ER Remastering
	- Point-In-Time Table level restores
	- View Logical Logs Backed Up by ON-Bar
	- ontape use of standard I/O
	- New ER commands to show statistics information

"Stober Drive is committed to Informix because it's the best database for us, because of reliability, because of scalability, <u>lack of administration</u> and the speed with which it returns our queries."
Bill Marshall, Operations Manager, Stober Drives, Inc.

Benefits of Manageability

The newer manageability features give database administrators (DBAs) a better view of the server. The benefits are that DBAs can be pro-active instead of reacting to situations. Better manageability means that DBAs can also provide their expertise to other section of the enterprise. One major benefit is then a more efficient use of the data server that results in optimized resource utilization. This optimization can translate in more efficient applications that provide better response time and greater scalability. This translates into a business advantage. This also results in savings that affect the business bottom line.

Application Development

Application development support includes more than database server features. It starts with the integration of IDS with other IBM Software Group (SWG) products. This include integration with Information Management products (Alphablox, DB2 Everyplace, Information Integrator, etc), WebSphere (App server, portal, etc), Tivoli (Data protection, monitoring), and Rational (Rational Application Developer, Rational Data Architect, etc).

Rational Application Developer (RAD) also includes a new fourth-generation language that was based in large part on Informix-4GL. This gives a migration path for 4GL customers that want to modernize both their development environment and their runtime environment. All this, while preserving their 4GL expertise. The RAD tool also includes a migration tool that automates most of the work to migrate 4GL code to the new EGL language.

The newest Client-Software Development Kit (CSDK) now includes a plug-in for Visual Studio .Net. This allows .Net developers to interact with the database without having to leave their IDE. Related to the client-SDK, we also have updates to the JDBC driver to track the evolution of the JDBC standard.

In addition to the benefits of the integration with the other SWG products and Visual Studio, IDS 10.0 includes features that can have a significant impact on software design and development. The following table summarizes the major features related to application development:

9.21 (and below)	9.40
- Extensibility features	- Iterator UDR in the FROM clause
- LVARCHAR (up to 2048 bytes)	- Sequence
- Re-entrant triggers	- ANSI join syntax
- SELECT trigger	- LVARCHAR up to 32000 bytes
- Long identifiers	- session-level non-default collation
- ability to retain update locks	- support for Unicode
9.30	- Microsoft Transaction Server
- Microsoft transaction server/XA support	tightly-coupled transaction support
- Display query plan without execution	- Support for FULL OUTER JOIN syntax
- NOT NULL properties used in	- Support for unions in subqueries of SELECT statements

outer join	- Trigger introspection
9.40	10.0
- INSTEAD OF trigger on views	- MQ DataBlade module
- SELECT order by column or expression not in projection list	- Truncate table
	- SELECT SKIP M FIRST N

Some of these features deserve further explanation. For others, please refer to the Informix documentation or ask your local IBM representative.

Extensibility features: These features allow you to better adapt the database to your environment. You could easily add functionality that would allow you to group your data by week of the year. You could go as far as having events that send messages to outside processes. For starter, please read the articles listed in the reference section to get an idea on how extensibility can change the way you look at application development.

LVARCHAR: This new data type gives you the ability to create variable length character strings of up to 32762 bytes. This could be a good replacement to the TEXT data type that some applications use for description or remark fields. The TEXT type is often use as a poor replacement to the VARCHAR type that can only accommodate 256 characters.

Long identifiers: You can now have table and column names and other identifiers of up to 128 characters. This gives more flexibility in naming but also makes it simpler to port applications from Oracle to IDS.

Sequences: IDS now support sequences by using the same syntax as in DB2 and Oracle. This is convenient when a unique identifier is needed for multiple tables and also for application porting to IDS.

Truncate table: The truncate table can quickly delete all the values in a table. This is more convenient than dropping and recreating a table.

Select skip M first N: This feature allows for skipping some of the resulting values and retrieving part of the resulting set. It is important for web applications that otherwise need to retrieve the whole result set.

All these features can have impacts in ease of development and porting.Moreover, these features could also have a significant impact in the performance of applications by providing better ways to approach a business problem.

Benefits of Application Development Features
The new application development capabilities of IDS 10.0 make it easier to adapt the database to the design of a solution. This translates into faster development, higher performance and less complexity. From this, we can also expect faster time to market and lower application maintenance costs.

IDS 10.0 also benefits from the performance improvements in the Client Software Development Kit (CSDK), the update to the JDBC driver for compliance with the latest specifications and the addition of the addition to development language support including PHP for web applications.

Other Considerations

IDS 10.0 also adds more flexibility in terms of packaging. It now comes in three versions: Express, Workgroup, and Enterprise. This packaging provides better match for different size customers or departments in large corporations. IDS 10.0 also supports additional platforms and updated versions of operating systems on the supported platforms.

Finally, the IBM migration toolkit version 1.4.7 supports migration from Oracle (8i, 9i, 10g) and Sybase Anywhere version 9 to IDS 10.0. This provides a path for consolidation for customers that have a mixed environment and want to take advantage of IDS' superior total cost of ownership.

Conclusion

IDS 10.0 has improved greatly in the areas of performance and scalability, reliability and availability, manageability, and application development. As noted above, each area provides significant benefits that result in a business advantages.

These benefits include better resource utilization that result in lower costs of operation, higher up time due to a set of features that also include better security, reduced administration overhead with the new manageability features, and easier software development with new server features and programming language support.

IDS 10.0 has significant benefits that should convince all current customers of its value. Adding to that the double digit new license growth seen this year (2006) and the inclusion of IDS in new leading edge application in fields such as Radio Frequency Identification (RFID), IDS 10.0 is the platform of choice for the future.

References

- IDS 7.x release notes
  //www.ibm.com/software/data/informix/pubs/library/notes/relnotes/servers2.html#ids7
- IDS 9.x release notes
  //www.ibm.com/software/data/informix/pubs/library/notes/relnotes/servers.html#ids9
- IDS 10.0 release notes
  //www.ibm.com/software/data/informix/pubs/library/notes/relnotes/servers.html#ids10
- IDS 10.0 Documentation
  //www-306.ibm.com/software/data/informix/pubs/library/ids_100.html

Success Stories

- Bayerwald sees higher productivity and lower costs with IBM Information On Demand. (Oct 2006)
  //www.ibm.com/software/success/cssdb.nsf/CS/LWIS-6UP2EH?OpenDocument&Site=dmmain&cty=en_us
- Harps Food finds support for rapid growth in IBM Informix and Gillani solution (Oct 2006)
  //www.ibm.com/software/success/cssdb.nsf/CS/LWIS-6UPTD5?OpenDocument&Site=dmmain&cty=en_us
- Everett School District bolsters its information management platform, improving data availability and reducing maintenance tasks (Sept 2006)
  //www.ibm.com/software/success/cssdb.nsf/CS/HSAZ-6U3TDA?OpenDocument&Site=dmmain&cty=en_us
- RealPlus accelerates online real estate listings service with IBM Informix Dynamic Server, Version 10 and Linux (Feb 2006)
  //www.ibm.com/software/success/cssdb.nsf/CS/HSAZ-6LYTRF?OpenDocument&Site=dmmain&cty=en_us
- Italian Foreign Ministry calls Linux solution from IBM its passport to success (Dec 2003)
  //www.ibm.com/software/success/cssdb.nsf/CS/TKNC-5U72GE?OpenDocument&Site=dmmain&cty=en_us
- TCH fuels its revenues with IBM Informix data management infrastructure (March 2003)
  //www.ibm.com/software/success/cssdb.nsf/CS/NAVO-5K734T?OpenDocument&Site=dmmain&cty=en_us

Extensibility examples

- Date Processing in Informix Dynamic Server
  //www.ibm.com/developerworks/db2/library/techarticle/dm-0510roy/
- Generating XML from IDS 9.x
  //www.ibm.com/developerworks/db2/zones/informix/library/techarticle/0302roy/0302roy2.html
- Event-driven fine-grained auditing for Informix Dynamic Server
  //www-106.ibm.com/developerworks/db2/library/techarticle/dm-0410roy
- Using GUIDs with IDS 9.x
  //www-106.ibm.com/developerworks/db2/library/techarticle/dm-0401roy/index.html