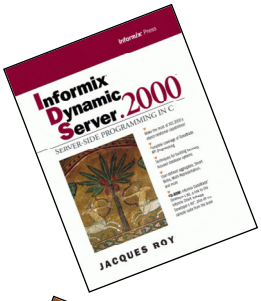
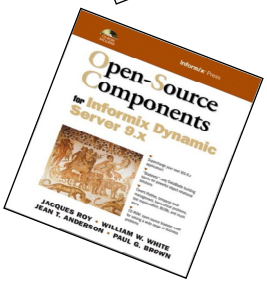
IDS 10.0 Deep Dive

Jacques Roy
Technical Lead
WW Information Management
jacquesr@us.ibm.com

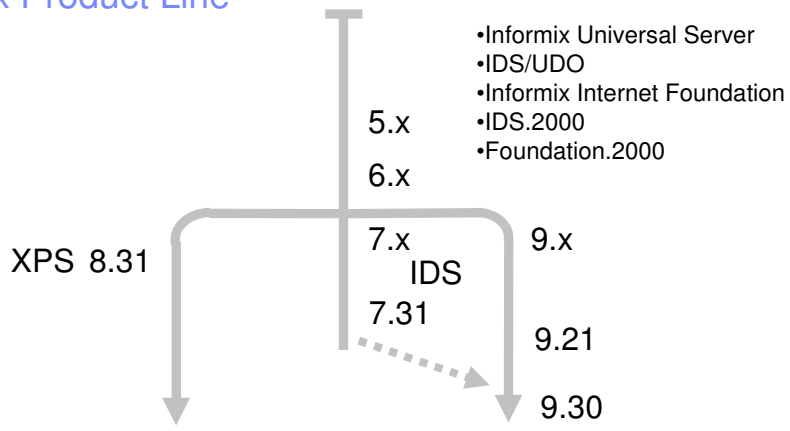
Who Am I?

- Informix/IBM 1995-Present:
  - Member of Informix ATG and Informix Labs
  - IDS Extensibility
  - WW DB Server Tech sales (IDS and DB2)
  - User Conferences
- Publications:
  - Informix Tech Notes
  - Informix Best Practices Books
  - IDS Extensibility Books
  - Developer Works Article on IDS and DB2 Extensibility
  - More…





Informix Product Line



Features Summary

- In 9.x (01/01/1997)
  Extensibility
  New data types (blob, boolean, clob, list, multiset, row, set)
- In 9.2x (03/30/2000)
  - Dynamic lock allocation
  - Fuzzy checkpoints
  - SQL Statement caching
  - Triggers on select statements
  - ON-Bar enhancements
  - Long identifiers
  - Parallel recovery
  - **From 7.3x:**
  - Attach/Detach fragments
  - Raw tables
  - Select first N
  - Restartable restore
- In 9.30 (09/17/2001)
  - Dynamic Log Allocation
  - Configurable Table Lock Mode
  - Explain without Execute
  - DELETE TABLE (no FROM)
  - REVOKE as USER
  - Temporary smartblobs
  - DataBlade API enhancements
  - Microsoft Transaction Server/XA support
  - **ER performance**, support for opaque UDTs, Blob/Clob

IDS 9.40

Improved Performance

- **Fractional LRU Min/Max**
- B-Tree Scanner
- Revised buffer priority mgmt
- Reduced code path
- **5-10% Faster than IDS 7.31**

Scalability

- **Up to 4TB Chunks**
- **Up To 128PB Instance**
- 2GB Utilities Limitation removed

Security

- **Encrypted Communications**
- **Plug-in Authentication Modules**
- No libraries in /usr/lib

Usability & Administration

- **HDR with ER**
- Onstat Enhancements
- Rename chunks during Restore
- Add chunks when 1st chunk full
- Restartable fast recovery
- TAPESIZE 0
- More DBSERVERALIASES
- New default ALARMPROGRAM
- Component installation
- AGS ServerStudio JE

Application Development

- **Free Spatial DataBlade**
- Improved CSDK
- Explain (avoid_execute)
- Retain Update Locks
- Non-Logging Tables (raw)
- ORDER BY values not in select-list
- Sequences
- Triggers on views (instead of)
- UNION in sub-queries
- Unicode support
- MANY MORE…

Latest Platform Support

… but best of all …

- 2 minute upgrade from 7.3

Where are we Today?

- Informix represents a significant portion of DB revenue
- Great features added in 9.40 and 10.0
  - IDS no limits (memory addressability, disk storage addressability)
- IDS activities
  - IDS 10 training, users group meetings, infobahns, chat with the lab, IDUG conference, tech conference, DB2 magazine articles.
- Recent increase in development resources
- .NET support (as of 9.4)
- New IDS PHP 5 and Python drivers
  - Needed for the open-source community
- New customers and new projects (ex: RFID)
  - IDS is growing!
- Ready to attack:
  - Migration toolkit (Oracle and Sybase) //www-306.ibm.com/software/data/db2/migration/mtk/
  - Competitive info //www.iiug.org/resources/articles/IDS10vsOracle10g.pdf

Topic List

ADMIN & USABILITY	PERFORMANCE	REPLICATION	AVAILABILITY	SECURITY	APPLICATIONS
Rename dbspace	Configurable Page Sizes ☆	DRAUTO	Online Index Build	PAM Authentication	JDBC 3.0 Support
Single-User mode	Memory Allocation to Non-PDQ Queries ☆	Replicate Templates	Faster Recovery w/ Fuzzies	Secure Environment Check	.NET Support
Multiple Fragments in one dbspace ☆	Dynamic OPTCOMPIND	Alter Table Support	Table Level Restore ☆	Datablade Registration Restrictions	ESQL/C to DB2
Tablespace Tablespace Management	External Optimizer Directives ☆	Fixing Corrupt Secondary Indexes ☆		Database Level Permissions	
SHMEM > 4G ☆		Replicate Resynch		Trigger Introspection	
HDR Setup w/ EBR				Denial-of-service attack	
ontape use of STDIO				Column-level encryption ☆	
Version Info					
Set Explain					
	B+Tree Scanners				Fixpack 3 ☆
Large Chunks/Files	Buffer Cache Manager				Fixpack 4 ☆

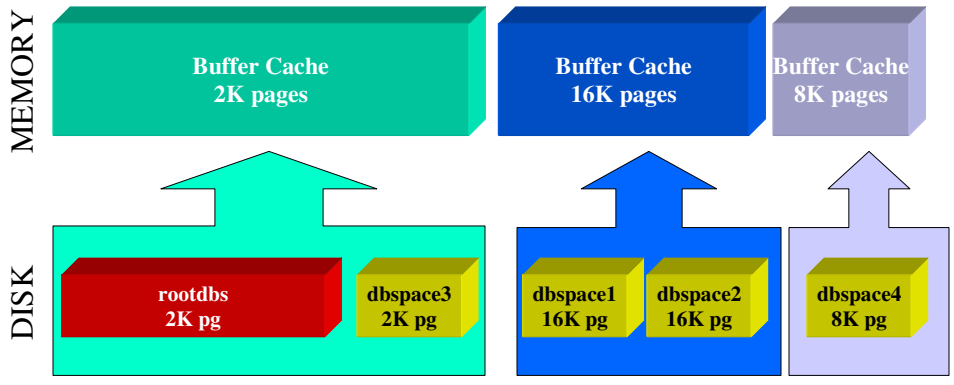
Configurable Page Sizes

- A feature that has been a “curiosity” to the customers.
- IDS XPS has a similar feature already
  - implemented differently – must define page size before building engine.
  - page size is the same for all structures.
- Similar to DB2
  - but not the same implementation.
  - and not for the same reason.

CPS Fun Facts

- can be defined at the *dbspace or buffer pool level*
- up to 16K page sizes; must be a multiple of 2K or 4K default page size for port
- new onconfig parameters added to manage these
- CPS requires “large chunk/file support” to be enabled (introduced in IDS 9.4)

Implementation – 2K Port



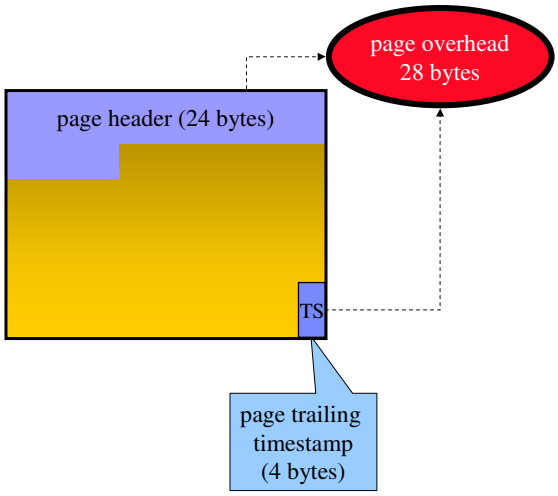
- **each port still has a “default page size” – 2K or 4K**
- **some structure must be in a dbspace with the default page size**

What did we solve with Configurable Pages?

- Objective #1: Space efficiency
  - larger pages up to **16K** bytes contiguous space
- Objective #2: Increased maximum key size & Unicode Support
  - longer keys up to **3000** bytes
- Objective #3: Access efficiency
  - less I/O operations for data and indices

First…some review on page/row overhead….

Page Overhead Basics

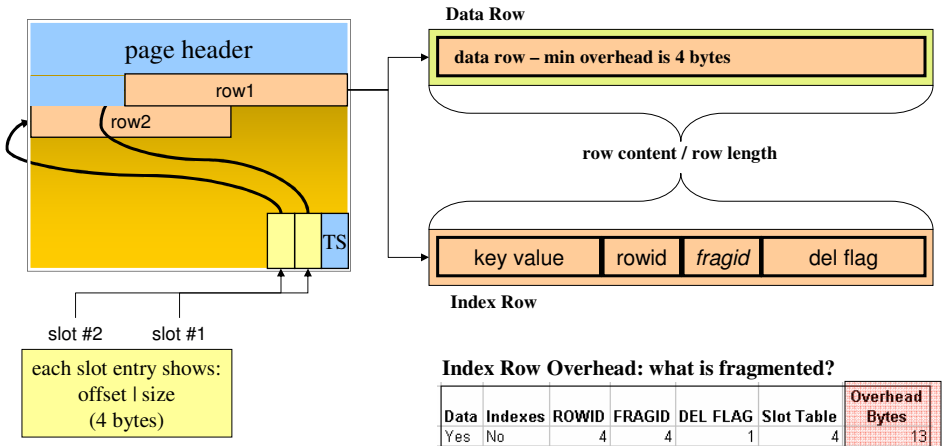


For example: a 2K page

page bytes	2048
page overhead	-28
available bytes	2020

Row Overhead Basics – Data/Index Pages

Data Row Overhead: minimum of 4 bytes for the slot table entry.



Index Row Overhead: what is fragmented?

Data	Indexes	ROWID	FRAGID	DEL FLAG	Slot Table	Overhead Bytes
Yes	No	4	4	1	4	13
No	Yes	4	0	1	4	9

Overhead Summary

- Page overhead 28 bytes
- Row overhead minimum - 4 bytes per row
  maximum – 13 bytes per row
- *With client tables approach 1B rows, this could be significant.*

Objective #1: Space Efficiency

So What About Space Efficiency?

- **Example**
- A row size of 1200 bytes:
  - 1 row fits on a 2k page (6k every 3 rows).
  - 3 rows fit on a 4k page, a savings of 33%
  .
- For thirty 1200-byte rows:
  - A 2k page size requires 60k.
  - A 4k page size would require only 40k.
  - A 6k page size requires just 36k, a 40% savings

For 1M rows, 4KB page saves 651MB!
6KB page saves 782MB!

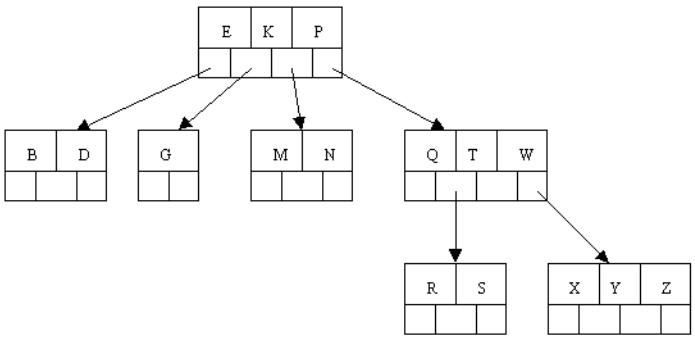
Objective #2: Larger Key Size & Unicode Support

Increased Maximum Key Size

- **Increased maximum key size - longer keys up to 3000 bytes**
  - Placing more keys on a page, we support longer keys without drastically increasing index level depth (maximum depth 20)
    - this is very significant for duplicate indexes with a high number of dups or volatility.
    - this won't change customers that are already building "wider indexes" – they can't span pages anyway.
  - The pre-10.0 key size limit was also a roadblock to utilizing the UNICODE character set, which caused some key values to exceed the max length.
- Maximum key size is ~page size / 5
  - 390 key for 2KB page size

B-Tree

- Stands for Balanced Tree
- Multiple types
  - B-Tree, **B+-Tree** (data records only in leaves), B*-Tree (tries to keep nodes 2/3 full)
- 2020/390 -> ~5.18



Proof – unique index (INT – 4 bytes)

2K dbspace

Index Usage Report for index tk1 on wisc_db:informix.tenktup1

Level	Total	Average No. Keys	Average Free Bytes
1	1	72	1160
2	72	138	214
Total	73	137	227

TBLspace Usage Report for wisc_db:informix.tenktup1

Type	Pages	Empty	Semi-Full
Free	2		
Bit-Map	1		
Index	73		
Data (Home)	0		
Total Pages	76		

16K dbspace

Index Usage Report for index tk2 on wisc_db:informix.tenktup2

Level	Total	Average No. Keys	Average Free Bytes
1	1	9	16252
2	9	1111	1911
Total	10	1000	3345

TBLspace Usage Report for wisc_db:informix.tenktup2

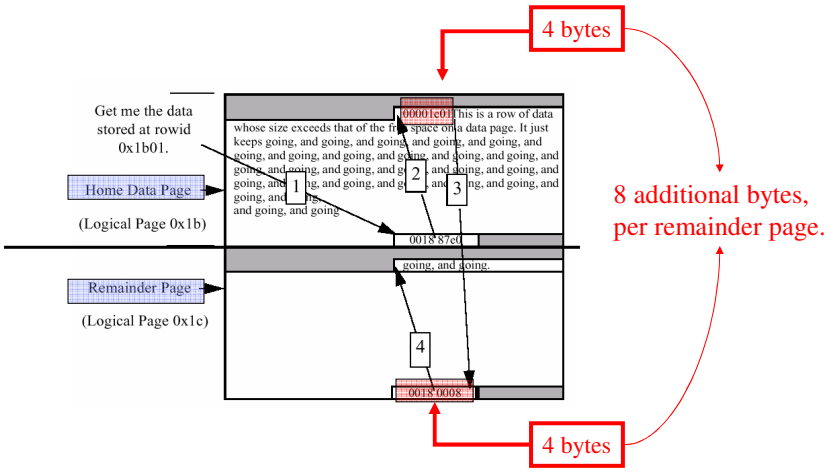
Type	Pages	Empty	Semi-Full
Free	1		
Bit-Map	1		
Index	10		
Data (Home)	0		
Total Pages	12		

Software Group – DB2 Information Management

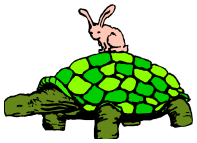
Objective #3: Access Inefficiencies

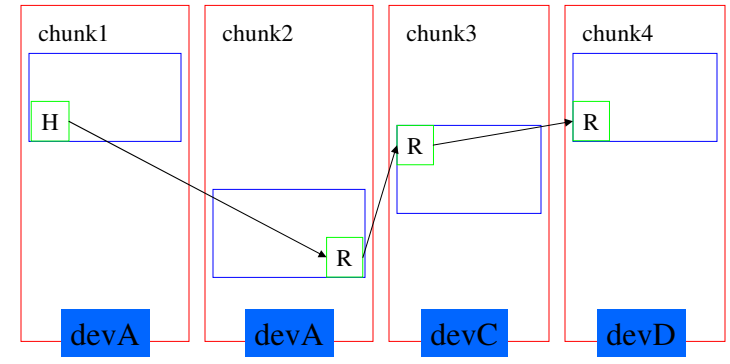
Access Inefficiencies : Forward Pointers

- Forward Pointers – data rows. On data pages - for rows larger than available page space on 2K or 4K ports, we utilize FORWARD POINTERS



Access Inefficiencies : Page Accesses





Number of Bufferpools

- Number of BUFFER pools is dependent upon default page size:
  - On a system with a 2K default page size, the maximum is 8
  - On a system with a 4K default page size, the maximum is 4
- ONCONFIG parameter BUFFERPOOL added to allow tuning of pools:

```
# The following parameters are related to the buffer pool
BUFFERPOOL      default,buffers=1000,lrus=8,lru_min_dirty=50.000000,lru_max_dirty=60.000000
BUFFERPOOL      size=2K,buffers=1000,lrus=8,lru_min_dirty=50.000000,lru_max_dirty=60.000000
BUFFERPOOL      size=16K,buffers=1000,lrus=8,lru_min_dirty=50.000000,lru_max_dirty=60.000000
~
~
```

REMEMBER:
There is one bufferpool *per page size.*

Example: Creating a 16K page dbspace

onspaces –c –d sparky –k 16 –p <path> -o offset -s <size>

```
informix:/export/home/informix> onstat -d

IBM Informix Dynamic Server Version 9.50.UC1B5   -- Single-User -- Up 12:34:39 --

Dbspaces
address  number   flags      fchunk   nchunks  pgsize   flags    owner    name
bd277d8  1        0x40001    1        1        2048     N  B     informix rootdbs
d08b3f0  2        0x40001    2        1        16384    N  B     informix sparky
 2 active, 2047 maximum
```

```
informix:/export/home/informix> onstat -b

IBM Informix Dynamic Server Version 10.00.UC1      -- On-Line -- Up

Buffers
address  userthread flgs pagenum          memaddr  nslots pgflgs

Buffer pool page size: 2048
 0 modified, 500000 total, 524288 hash buckets, 2048 buffer size

Buffer pool page size: 16384
 0 modified, 1000 total, 1024 hash buckets, 16384 buffer size
```

Page Concepts

Changing:

- home data pages
- remainder pages
- partition freemap pages
- partition partition pages
- chunk free list pages
- index pages
- partition blob pages - partition blobs will use the same page size as the large page size defined for the dbspace they reside in.

Not changing:

- maximum number of pages per partition still 16,777,216.
- maximum number of rows per page remains at 255.
- maximum number of parts per key remains 16.
- text or byte types in a blobspace are not affected.
- smartblobs are not affected.
- user-defined external spaces are not affected.
- r-tree indexes must be stored in a dbspace with the default page size.
- the rootdbs must be the default page size.
- the physical and logical log dbspaces must be the default page size.
- dynamically created logs must be in a default page size dbspace.

Page Structure

- **only one page header per large page.**

Page Dump Example – a data page from partition 2097204

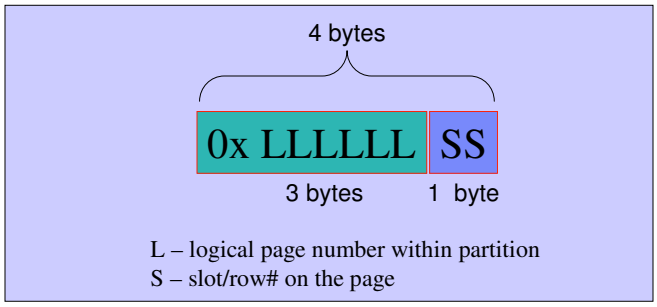
oncheck -pp 2097204 1

```
addr              stamp    chksum nslots flag type        frptr frcnt next     prev
2:2816            85712    45d3   28     801  DATA        3776  12492 0        0
        slot ptr   len   flg
        1    24    134   0
        2    158   134   0
        3    292   134   0
        4    426   134   0
```

```
28 rows * 134   =        3752      # row bytes
28 rows * 4     =         112      # slot bytes
page hdr/trl TS =          28      # page overhead

3752 + 112 + 28 = 3892             # used bytes

16384 – 3892 =  12492              # free count (bytes)
```

Row Size / Number of Rows Considerations

- Primary purpose for CPS is to accommodate large rows (rows larger than the default page size).
- Be careful!
  - the “rows per data page limit” of 255 is STILL enforced with 10.0.
  - *regardless of row size, the max rows a data page can house is 255.*
  - this limitation is due to the ROWID format

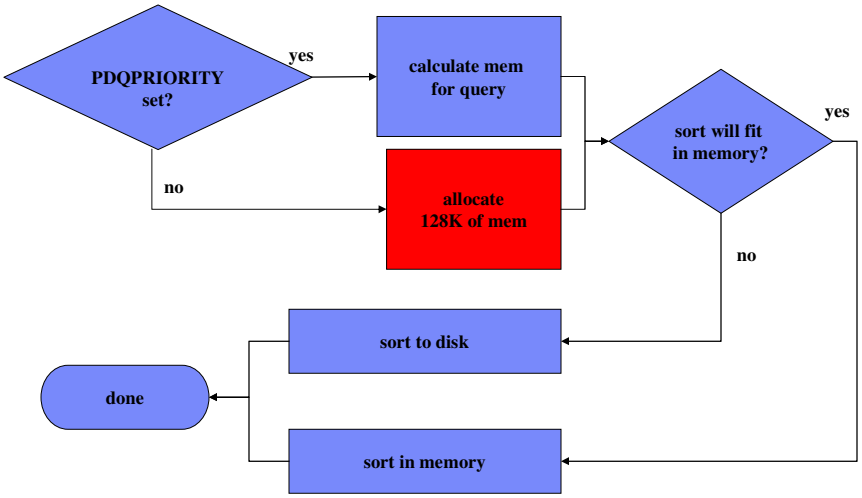


onstat –g buf

```
informix:/export/home/informix> onstat -g buf |more

IBM Informix Dynamic Server Version 9.50.UC1B5   -- On-Line -- Up 10:14:56 -- 10137

Profile

Buffer pool page size: 2048
dskreads   pagreads   bufreads   %cached dskwrits   pagwrits   bufwrits   %cached
1346       7542       6244       78.44   527        7542       319         0.00
bufwrits_sinceckpt  bufwaits   ovbuff     flushes
0                   2          0          9

Fg Writes     LRU Writes    Chunk Writes
0             0             124


Buffer pool page size: 16384
dskreads   pagreads   bufreads   %cached dskwrits   pagwrits   bufwrits   %cached
164        1312       30379      99.46   492        1312       10972       95.52
bufwrits_sinceckpt  bufwaits   ovbuff     flushes
0                   0          0          5

Fg Writes     LRU Writes    Chunk Writes
0             0             492
```

Page 1

Performance : Non-PDQ queries - a problem….



Performance : Memory Allocation for non-PDQ Queries

- This feature was first available in 9.40.xC4.
- You can specify how much memory is allocated to non-PDQ queries.
  - The default of 128K can be insufficient for queries that specify ORDER BY, GROUP BY, hash joins, or other memory-intensive options.
- Use the new configuration parameter, DS_NONPDQ_QUERY_MEM, to specify more memory than the 128K that is allocated to non-PDQ queries by default.
- The DS_NONPDQ_QUERY_MEM value is calculated:
  - during database server initialization based on the calculated DS_TOTAL_MEMORY value.
    or
  - when an **onmode [-wf | -wm] DS_NONPDQ_QUERY_MEM=<value>**

  -wf….changes the onconfig parameter
  -wm...changes the session setting.
- Memory used by a statement: onstat –g stm {session_id} (heapsz field)

DS_NONPDQ_QUERY_MEM examples

Testing values set when initializing engine:

SMVIRTSIZE	DS_TOTAL_MEMORY	DS_NONPDQ_QUERY_MEM	Allocated (Kb)	Ratio
2,000,000	-	1,000,000	256	min
2,000,000	1,000,000	1,000,000	250,000	0.25
2,000,000	2,000,000.00	1,000,000	500,000	0.25
2,000,000	2,000,000.00	4,000,000	500,000	0.25

Changing the onconfig parameter: **onmode –wf DS_NONPDQ_QUERY_MEM**

```
informix:/export/home/informix> onmode -wf DS_NONPDQ_QUERY_MEM=2000000
06:27:01  Must be >= 128 and <= 0.25 * DS_TOTAL_MEMORY. Or blank for default value.
```

```
informix:/export/home/informix> onmode -wf DS_NONPDQ_QUERY_MEM=500000
Value of DS_NONPDQ_QUERY_MEM has been changed to 500000.
```

** DS_NONPDQ_QUERY_MEM can also be changed just for the session*:
onmode –wm DS_NONPDQ_QUERY_MEM=<value>

Page 1

Performance : Storing/Applying External Directives

- You can create, save, and reuse external optimizer directives.
- External optimizer directives are useful when it is not feasible to rewrite a query for a short-term solution to a problem
  - for example, when a query starts to perform poorly.
- This feature is implemented as a new SQL statement, SAVE EXTERNAL DIRECTIVES
  - creates and registers external optimizer directives in a new **sysdirectives** table of the system catalog.
  - Use the new **IFX_EXTDIRECTIVES** environment variable or the **IFX_EXTDIRECTIVES** configuration parameter to enable this feature.

Usage

SAVE EXTERNAL DIRECTIVES *directives* [ACTIVE/INACTIVE/TEST ONLY] FOR Select/* INDEX(table1 , index1)*/ col1 , col2 From table1, table 2 Where table1.col1+table2.col1

- You must include one of the ACTIVE, INACTIVE, or TEST ONLY keyword options to enable, disable, or restrict the scope of external directives:
  - If external directives are enabled, the ACTIVE keyword applies the list of directives to any subsequent query that matches the *query* string.
  - The INACTIVE keyword causes the engine to ignore the directive. (It is associated with the query in **sysdirectives**, but it is dormant, with no effect.)
  - If external directives are enabled, the TEST ONLY keywords apply the directives only to matching queries that the DBA or user **informix** executes. Queries by any other users cannot use TEST ONLY external directives.

Example

This associates AVOID_INDEX and FULL directives with the specified query.

SAVE EXTERNAL DIRECTIVES ACTIVE FOR **/*+ AVOID_INDEX (table1 index1)*/** , **/*+ FULL(table1) */**

```
SELECT /*+ INDEX( table1 index1 ) */ col1, col2 FROM table1, table2
WHERE table1.col1 = table2.col1
```

The inline INDEX directive is ignored by the optimizer when the external directives are applied to a query that matches the SELECT statement.

- this is a “negative directive” due to the use of AVOID.
- a negative directive gives the optimizer choices when it chooses the path for the query plan.

Enabling External Directives for a Session

- External directives are ignored if the EXT_DIRECTIVES parameter is set to 0 in the ONCONFIG file. In addition, the client system can disable this feature for its current session by setting the **IFX_EXTDIRECTIVES** environment variable to 0.
- The following table shows whether external directives are disabled (OFF) or enabled (ON) for various combinations of valid **IFX_EXTDIRECTIVES** settings on the client system and valid EXT_DIRECTIVES settings on Dynamic Server:

	EXT_DIRECTIVES = 0	EXT_DIRECTIVES = 1	EXT_DIRECTIVES = 2
IFX_EXTDIRECTIVES Not Set	OFF	OFF	ON
IFX_EXTDIRECTIVES = **1**	OFF	ON	ON
IFX_EXTDIRECTIVES = **0**	OFF	OFF	OFF

Example

Query execution before saving external directives.

```
QUERY:
------
select --+EXPLAIN
        * from customer c, orders o
        where c.customer_num = o.customer_num

DIRECTIVES FOLLOWED:
EXPLAIN
DIRECTIVES NOT FOLLOWED:

Estimated Cost: 8
Estimated # of Rows Returned: 22
  1) informix.o: SEQUENTIAL SCAN
  2) informix.c: INDEX PATH
     (1) Index Keys: customer_num   (Serial, fragments: ALL)
         Lower Index Filter: informix.c.customer_num = informix.o.customer_num
NESTED LOOP JOIN
```

Expected access methods

Expected Join Algorithm

Save the External Directive

```
SAVE EXTERNAL DIRECTIVES
 {+USE_HASH(customer /BUILD), USE_HASH(orders /PROBE)}
ACTIVE FOR
        select --+EXPLAIN
        * from customer c, orders o
        where c.customer_num = o.customer_num;

External directives saved.
```

Attempting to force a HASH JOIN versus NESTED LOOP JOIN

select * from sysdirectives; -- to show the stored directive

```
id              2
query
select --+EXPLAIN
        * from customer c, orders o
        where c.customer_num = o.customer_num
directive
USE_HASH(customer /BUILD), USE_HASH(orders /PROBE)
directivecode   <BYTE value>
active          1
hashcode        862580839
```


Results when using External Directives

```
QUERY:
------
select --+EXPLAIN
* from customer c, orders o
where c.customer_num = o.customer_num

External Directives in effect.
DIRECTIVES FOLLOWED:
USE_HASH ( customer/BUILD )
USE_HASH ( orders/PROBE )
DIRECTIVES NOT FOLLOWED:

Estimated Cost: 14
Estimated # of Rows Returned: 22

  1) informix.c: SEQUENTIAL SCAN

  2) informix.o: SEQUENTIAL SCAN


DYNAMIC HASH JOIN (Build Outer)
    Dynamic Hash Filters: informix.c.custome
```

Hash join enforced

WARNING….WARNING…

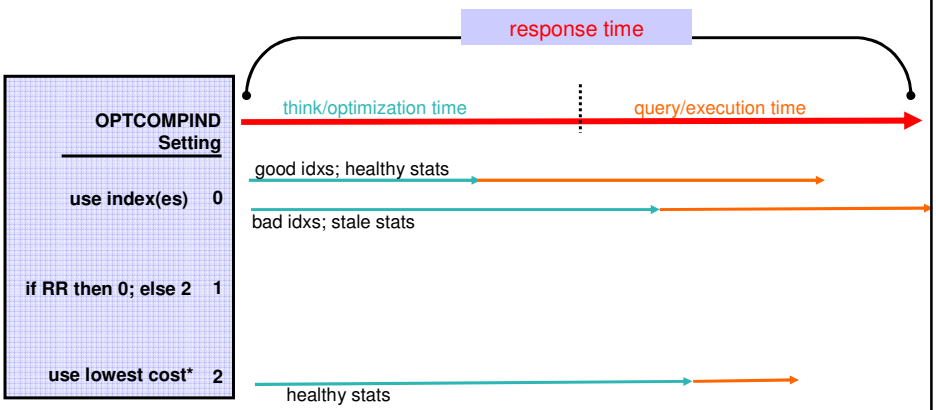

```
QUERY:
------
select --+EXPLAIN
*  from customer c, orders o
where c.customer_num = o.customer_num

DIRECTIVES FOLLOWED:
EXPLAIN
DIRECTIVES NOT FOLLOWED:

Estimated Cost: 8
Estimated # of Rows Returned: 22

  1) informix.o: SEQUENTIAL SCAN

  2) informix.c: INDEX PATH

    (1) Index Keys: customer_num   (Seri
        Lower Index Filter: informix.c.c
NESTED LOOP JOIN
```

WHY wasn't the Saved Directive applied this time?

Page 1

Performance: OPTCOMPIND

A Review



* cost = I/O + (cpu *.03)

Performance : Dynamic OPTCOMPIND

- You can use SET ENVIRONMENT OPTCOMPIND to set **OPTCOMPIND** environment variable dynamically for the current session.
- The value that you enter using this statement takes precedence over the current setting specified in the ONCONFIG file.
- The default setting of the **OPTCOMPIND** environment variable is restored when your current session terminates.
- No other user sessions are affected by SET ENVIRONMENT OPTCOMPIND statements that you execute.

Page 1

Security Enhancements: Column Level Encryption

Highlights

- Built-in SQL encryption functions to support data encryption
- Latest cryptographic standards
- 128 bit AES and Triple-DES
- Passwords up to 128 characters
- View/trigger/SPL support

Security Enhancements: Column Level Encryption

- Can be used on the following datatypes:
  - CHAR
  - NCHAR
  - VARCHAR
  - NVARCHAR
  - LVARCHAR
  - BLOB
  - CLOB
- Usage examples
  - INSERT INTO table VALUES (1, ENCRYPT_AES(data, password));
  - UPDATE table SET column = ENCRYPT_TDES(data, password) WHERE …;
  - SELECT DECRYPT_CHAR(column, password) FROM table;
  - EXECUTE FUNCTION ENCRYPT_AES(data, password, hint);

Security Enhancements: Column Level Encryption

- Example – table creation / data insert
  - Create table

```
create table customer
    (
    customer_num                serial(101),
    fname                       char(43), -- encrypted char(15)
    lname                       char(43), -- encrypted char(15)
    company                     char(87), -- encrypted char(40)
    address1                    char(67), -- encrypted char(30)
    address2                    char(67), -- encrypted char(30)
    city                        char(15),
    state                       char(2),
    zipcode                     char(5),
    phone                       char(67), -- encrypted char(18)
    primary key (customer_num)
    );
```

  - Insert data

```
set encryption password "my password";
insert into customer values (101, encrypt_aes("Ludwig"), encrypt_aes("Pauli"),
    encrypt_aes("All Sports Supplies"), encrypt_aes("213 Erstwild Court"),
     encrypt_aes(""), "Sunnyvale", "CA", "94086", encrypt_tdes("408-789-8075"));
```

Security Enhancements: Column Level Encryption

- Example (Query)
  - Query

```
SELECT * FROM customer WHERE customer_num = 101
customer_num    101
fname           0tkn/AAAAEAndSyVxZpHZmA2/mCoJ6uZUHDSQ5I5u3l
lname           0V6j/AAAAEApupW+hft1mjw2CorFt7P9oWjO6cljVzs
company         0Mc7/AAAAIAJbjI7Ul6N4oHMgIfh5wGo3559mnm8dIwznU2C+eivERQYUKse1WhoQ==
address1        0qVb/AAAAIAXoyVeIIDxAWV8MqsX8mTdHDWCpG6A1bjGGgPdYioigDAz3/OdDinHw==
address2        0AP//AAAAAAe+M+wFglsPA=
city            Sunnyvale
state           CA
zipcode         94086
phone           1p1j/AAAAEAA1viBow2uu+7fLLJbEtuMucwiv9q0Mmx

SELECT customer_num, DECRYPT_CHAR(fname, 'my password') as fname,
      DECRYPT_CHAR(lname, 'my password') as lname FROM customer
      WHERE customer_num = 101
customer_num  101
fname                 Ludwig
lname                 Pauli
```


Security Enhancements: Column Level Encryption

- Example (view)
  - Create a view

```
SET ENCRYPTION PASSWORD null;
CREATE VIEW custview (customer_num, firstname, lastname, company, city)
   AS SELECT customer_num, DECRYPT_CHAR(fname),
        DECRYPT_CHAR(lname), DECRYPT_CHAR(company), city
   FROM customer
```

  - Query from view

```
SET ENCRYPTION PASSWORD 'my password';
SELECT * FROM custview;
      customer_num    101
      firstname       Ludwig
      lastname        Pauli
      company         All Sports Supplies
      city            Sunnyvale
```

Page Dump Example – Physical Storage

```
drop table duckman;
create table duckman (col1 char(200));

set encryption password "12 Monkeys";
insert into duckman values ("Not encrypted");
insert into duckman values (ENCRYPT_TDES("Encrypted"));
```

```
informix:/export/home/informix> oncheck -pp 2097234 1
addr             stamp    chksum nslots flag type          frptr frcnt next
2:3840           104298   9869   2      801  DATA          424   15948 0
        slot ptr   len   flg
        1    24    200   0
        2    224   200   0
slot   1:
    0: 4e 6f 74 20 65 6e 63 72 79 70 74 65 64 20 20 20   Not encrypted
   16: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   32: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   48: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   64: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   80: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   96: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  112: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  128: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  144: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  160: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  176: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
  192: 20 20 20 20 20 20 20 20                         ........
slot   2:
    0: 31 74 45 76 2f 41 41 41 41 45 41 59 35 51 79 54   1tEv/AAAAEAY5QyT
   16: 58 4e 70 58 47 74 50 78 56 4b 6a 6c 78 43 2f 73   XNpXGtPxVKjlxC/s
   32: 67 76 53 67 7a 35 58 4a 6e 6b 70 20 20 20 20 20   gvSgz5XJnkp
   48: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
   64: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
```


Passwords

When you set a password, IDS transfers the password and any hint to a 128-bit key that is used to encrypt the password and hint.

Passwords and hints are not stored as clear text:

- The key is a time-based random value per instance.
- The engine initializes the key when it starts.
- The key is destroyed when the database server shuts down.

Password Visibility (or not)

```
SELECT DECRYPT_CHAR(col1, "12 monkeys") FROM duckman;
```

```
$ID:$PWD> onstat -g sql 26

IBM Informix Dynamic Server Version 10.00.UC1      -- Sin

Sess  SQL             Current            Iso Lock        S
Id    Stmt type       Database           Lvl Mode        E
26    -               skippy             CR  Not Wait    0

Last parsed SQL statement :
  select DECRYPT_CHAR(col1,"xxxxxxxxxx") from duckman
```


IMPORTANT

- If the *column size is smaller than the returned data size* from ENCRYPT/DECRYPT functions:
  - the encrypted data is truncated when it is inserted
  - it is ***not possible to decrypt the data***

  (IDS Reference Manual documentation)

Example of Truncated Data

```
drop table too_short;
create table too_short(col1 char(20));

insert into too_short values (ENCRYPT_AES("Mark","IDS Lives"));

select * from too_short;

select DECRYPT_CHAR(col1,"IDS Lives") from too_short;


col1

OS7T/AAAAEA5gNyD/aL4

 26005: The encrypted data is wrong or corrupted.

$ID:$PWD> finderr 26005
-26005  The encrypted data is wrong or corrupted.

The data must be a result of an IDS encryption function. The decryption
may have failed because the data was not encrypted, was corrupted, or
truncated (e.g. ALTER TABLE operation) since encryption, or was
encrypted with an algorithm not supported by this version.
```


Encryption Space Estimate

Base64 encoding stores 6 bits of data as 8 bits: (4N + 4) / 3

A: Plain text data size
B: Block size (8 bytes for TDES)
C: Hint size
D: IV size (8 bytes)
E: Header size for base64 encoding (11)

Size = (4 * ((B * ((A/B + 1) + (C/B + 1))) + D) / 3) + E

For BLOB and CLOB:
TDES: N + HintSize + 24
AES : N + hitSize + 32

IDS 10.0 SQL Syntax page 4-99

Encryption VPs

- IDS includes an Encrypt Virtual Processor.
- If the encrypt option of the VPCLASS parameter is not defined in the ONCONFIG configuration file, the database server starts one Encrypt VP the first time that any encryption or decryption functions defined for column-level encryption are called.
  - You can define multiple Encrypt VPs if necessary to decrease the time needed to start the database server.
- When the database server is in online mode, you can use the **onmode -p** command to add or drop Encrypt VPs.

onmode -p 4 encrypt	# to add 4 Encrypt VPs
onmode -p -3 encrypt	# to drop 3 Encrypt VPs

Encryption VPs

1. After engine initialization:

```
Individual virtual processors:
 vp     pid        class        usercpu    syscpu     total
 1      1287       cpu          0.26       0.34       0.60
 2      1288       adm          0.00       0.00       0.00
 3      1289       cpu          3.93       0.03       3.96
 4      1290       cpu          0.00       0.01       0.01
 5      1291       cpu          0.00       0.00       0.00
 6      1292       lio          0.00       0.00       0.00
 7      1293       pio          0.00       0.00       0.00
 8      1294       aio          0.00       0.00       0.00
 9      1295       msc          0.00       0.00       0.00
 10     1296       aio          0.00       0.00       0.00
 11     1297       tli          0.00       0.00       0.00
                   tot          4.19       0.38       4.57
```

2. insert into blah values (ENCRYPT_AES("something","gabbel ratchet", "genesis"))

```
Individual virtual processors:
 vp     pid        class        usercpu    syscpu     total
 1      1287       cpu          0.28       0.35       0.63
 2      1288       adm          0.00       0.00       0.00
 3      1289       cpu          3.93       0.04       3.97
 4      1290       cpu          0.04       0.01       0.05
 5      1291       cpu          0.00       0.00       0.00
 6      1292       lio          0.00       0.00       0.00
 7      1293       pio          0.00       0.00       0.00
 8      1294       aio          0.00       0.00       0.00
 9      1295       msc          0.00       0.00       0.00
 10     1296       aio          0.00       0.00       0.00
 11     1297       tli          0.00       0.00       0.00
 12     1300       encrypt      0.00       0.00       0.00
                   tot          4.25       0.40       4.65
```

Page 1

Security: Secure Environment Check

- Server utilities on UNIX now check if the environment is secure by testing for the following:
  - The permissions on **$INFORMIXDIR** and some directories under it are correct. For each directory, check that the directory exists, is owned by user **informix** and the correct group, and that its permissions do not include write permissions for the group or other users.
  - The permissions on the ONCONFIG file are correct. The file must belong to the DBSA group. If the DBSA group is group informix (default), then the ONCONFIG file should be owned by user **informix** too; otherwise, the ownership is not constrained. The file must not have write permissions for others.
  - The permissions on the **sqlhosts** file are correct. Under a default configuration, the **sqlhosts** file is **$INFORMIXDIR/etc/sqlhosts**; the owner should be user **informix**, the group should be either the **informix** group or the DBSA group, and there should be no public write permissions. If the file is specified by setting the **INFORMIXSQLHOSTS** environment variable, then the owner and group are not checked, but public write permissions are not permitted.
  - The length of both the filenames **$INFORMIXDIR/etc/onconfig.std** and **$INFORMIXDIR/etc/$ONCONFIG** must be less than 256 characters.

Page 1

Preventing Denial-of-Service Attacks

- Multiple listening threads
- Limit on availability of listeners for incomplete connections
- Default for incomplete connections timeout reduced
  - From 60 seconds to 10 seconds
- Default Maximum number of incomplete connections: 1024
- Two new configuration parameters
  - LISTEN_TIMEOUT
  - MAX_INCOMPLETE_CONNECTION
  - Can be changed with onmode

Page 1

Security: Trigger Introspection

- You can create user-defined routines that are invoked in trigger action statements
  - to obtain information about the triggers, triggering tables, views, statements, and the values of rows involved in the trigger actions.
  - Using the new DataBlade API routines, you can write a general purpose user-defined routine that can you can use to audit any table and any trigger event

Page 1

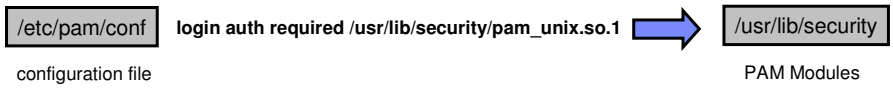
Security: Restricting Registration of DataBlade Modules and UDRs

- The DBSA (Database Server Administrator) can use a new built-in role, called EXTEND, to specify which users can register UDRs that include the EXTERNAL NAME clause.
- User-defined routines use shared-object files that are external to the database server and that could potentially contain harmful code.
- The DBSA can disable this feature by setting to “off” a new IDX_EXTEND_ROLE configuration parameter. This feature is intended to improve security and to control accessibility.

Page 1

Security: PAM Authentication

- PAM – Pluggable Authentication Modules
  - It is a standardized system for allowing the OSA (Operating System Admininistrator) to configure how authentication is done.
  - Allows OSA to configure authentication methods.
  - Available on Linux, AIX, Solaris, HP-UX and others.
  - configured at the Operating System level.
  - APIs to write shared object (.so) at wws.sun.com/software/solaris/pam
  - PAM supports challenge-response protocols:
    - In response to initial authentication request,
    - PAM issues a challenge,
    - And waits for response from application.



Page 1

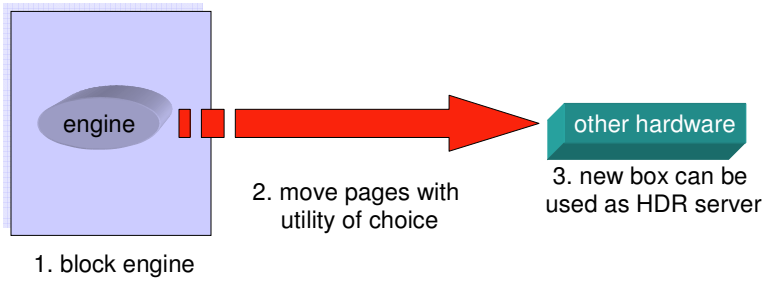
Administration: HDR Setup with EBR

HDR – High Availability Data Replication

- entire engine replicated.
- used for hot backups typically.
- used also for not-realtime report server.

EBR – External Backup and Restore

- a “non-IDS” backup to media of DBA’s choice
- added many releases ago to allow clients to, for example, “break off a mirror” for the archive/backup.



Page 1

Administration : **Single User Mode**

Overview

- a mode intermediate between Quiescent mode & Online mode
- allows only the user 'informix' to connect and do any required maintenance

Utility Enhancements

- **oninit -j** … brings the engine from *offline* to *single-user mode*.
- **onmode -j** … brings the engine from *online* to *single-user mode*.

```
informix:/export/home/informix> onmode -j
This will change mode to single user. Only user informix can connect
in this mode.
Do you wish to continue (y/n)? y

All threads which are not owned by user informix will be killed.
Do you wish to continue (y/n)? y
informix:/export/home/informix> onstat -

IBM Informix Dynamic Server Version 9.50.UC1B5   -- Single-User -- Up

informix:/export/home/informix>
```

Page 1

Administration: Renaming Dbspaces

Description

The ability to change the name of a previously defined dbspace.

The Problem

- Existing customers who wish to reorganize their data are moving their data to a new dbspace, then reloading the data back into the original dbspace to regain the original dbspace name. The last reload step could be avoided by using a rename dbspace option. Helps in Recycling of Dbspaces.
- Time consuming operations as reorganizing the data in an existing dbspace can benefit from this feature.

- **The rename dbspace operation only changes the dbspace name; it does not reorganize data**

Administration : Renaming Dbspaces

9.40.UC3

- Feature was first introduced with limitations:
  - You cannot rename blobspaces, sbspaces, temporary, or external spaces.
  - cannot rename dbspaces that are referred to by the following configuration parameters:

    DBSPACETEMP, CDR_DBSPACE
    SBSPACENAME, SBSPACETEMP
    SYSSBSPACENEAME, CDR_QHDR_DBSPACE
    CDR_QDATA_SBSPACE,

  - Cannot rename dbspaces if you are using High-Availability Data Replication.
  - Cannot rename dbspaces when Enterprise Replication is active.

Administration : Renaming Dbspaces

v10.0 Restrictions:

- Rename can not be done on critical spaces (i.e.) root dbspace, space containing physical log or logical logs.
- A dbspace with down chunks can not be renamed.
- Rename of spaces cannot be done with onmonitor.

Administration : Renaming Dbspaces

- A Level 0 archive of the renamed space and root dbspace need to be taken after renaming.
- Rename done on HDR primary will propagate to the secondary.

Rename Dbspace Example

```
informix:/export/home/informix> onspaces -c -d skippy -k 16 -p /dev/rdsk/stu414B -o 0 -s 100000
Verifying physical disk space, please wait ...
Cannot build a new Space.
ISAM error: Cannot add dbspace of big page when Large Chunk support is disabled.

informix:/export/home/informix> onmode -BC 1
This command will enable creation of large chunks.
** WARNING **  This action cannot be undone.
** WARNING **  A level 0 archive of Root DBSpace will need to be done.
Do you wish to continue (y/n)? y

Expanded chunk capacity mode: enabled
informix:/export/home/informix> onmode -BC 2
This command will cause all chunks to be written in the new (big) format.
** WARNING **  This action cannot be undone.
** WARNING **  A level 0 archive of Root DBSpace will need to be done.
Do you wish to continue (y/n)? y

Expanded chunk capacity mode: always
informix:/export/home/informix> onspaces -c -d skippy -k 16 -p /dev/rdsk/stu414B -o 0 -s 100000
Verifying physical disk space, please wait ...
Space successfully added.

** WARNING **  A level 0 archive of Root DBSpace will need to be done.
informix:/export/home/informix>
```

Administration : Rename Dbspace

```
informix:/export/home/informix> onstat -d

IBM Informix Dynamic Server Version 9.50.UC1B5   -- On-Line -- Up 00:02:17 -- 1013760 Kb

Dbspaces
address  number   flags      fchunk   nchunks  pgsize   flags    owner    name
ad277d8  1        0x60001    1        1        2048     N  B     informix rootdbs
c374410  2        0x60001    2        1        16384    N  B     informix skippy
 2 active, 2047 maximum

Chunks
address  chunk/dbs  offset     size       free       bpages     flags pathname
ad27928  1     1    0          50000      40212                 PO-B  /dev/rdsk/stu416A
c39e1a0  2     2    0          6250       6197                  PO-B  /dev/rdsk/stu414B
 2 active, 32766 maximum

NOTE: The values in the "size" and "free" columns for DBspace chunks are
      displayed in terms of "pgsize" of the DBspace to which they belong.

Expanded chunk capacity mode: always

informix:/export/home/informix>
```


Rename Dbspace

```
informix:/export/home/informix> onmode -s

This will perform a GRACEFUL SHUTDOWN -
Do you wish to continue (y/n)? y
informix:/export/home/informix> onspaces -ren skippy -n sparky
Rename of Space completed successfully.

** WARNING **  A level 0 archive of Root DBSpace and the renamed
Space need to be done.
informix:/export/home/informix> onstat -d

IBM Informix Dynamic Server Version 9.50.UC1B5   -- Quiescent -- Up 00:04:07 -- 101376

Dbspaces
address  number   flags      fchunk   nchunks  pgsize   flags    owner    name
ad277d8  1        0x60001    1        1        2048     N  B     informix rootdbs
c374410  2        0x60001    2        1        16384    N  B     informix sparky
 2 active, 2047 maximum
```

Page 1

Administration : Per Database Permissions

- You can create a default role and assign that role to individual users or to PUBLIC on a per-database level.
- Each user who is assigned to a default role receives the privileges of that role as well as whatever other privileges are granted to the user individually.
- The syntax of the GRANT, REVOKE, and SET ROLE statements support this feature.

Page 1

Administration: ontape use of standard I/O

- ontape can now use standard I/O instead of a tape device or disk file.
- Specifying stdout or stdin allows **ontape** to use pipes or programs for archives and restores.
  - For example, you can use compression to save media space, use cloning to duplicate the archive for safety reasons, or restore the data onto another server instance.
  - This feature is especially efficient for setting up High-Data Availability Replication by restoring the data to the secondary server while skipping the intermediary step of saving the data to a file or disk.

Page 1

ALRM_ALL_EVENTS

- Configuration parameter
- Indicate to send ALL events to the alarm program
  - Instead of severity > 1
- Severity:
  - 1: Not Noteworthy (ex: data change in the message log)
  - 2: Information (ex: log backup completed)
  - 3: Attention (ex: one chunk of a mirrored pair goes down)
  - 4: Emergency (ex: assert failure, oncheck reports data corruption)
  - 5: Fatal

Page 1

Version Information

- New option for all utilities: -version
- More information than –V
- Ex:

```
onstat –V
IBM Informix Dynamic Server Version 10.00.TC2E    Software Serial Number
AAA#B000000

onstat –version
Program Name:   onstat
Build Version:  10.00.TC2E
Build Number:   N049
Build Host:     HAL08
Build OS:       Windows_NT 5
Build Date:     Fri Apr 8 18:15:55 CDT 2005
GLS Version:    glslib-4.00.TC5
```

Page 1

Administration : Managing the Tblspace Tblspace

- Overview
  - This feature changes the behavior of the sizing of the tblspace tblspace. It also changes the behavior of extending the tblspace tblspace.
- Description
  - A DBA will now be allowed to specify the first and next extent sizes for the tblspace tblspace.
  - When extending the tblspace tblspace preferences will be made when choosing the chunk for the next extent.

Administration : Managing the Tblspace Tblspace

- Tblspace == partition
- TBLspace TBLspace
  - Special table that tracks other tables within a dbspace.
  - Every dbspace has its own tblspace tblspace that tracks tables (partitions) within its own dbspace.
  - There is only **one** tblspace tblspace per dbspace. But as with any tblspace it can have multiple extents. (see next slide)
  - Partnum is **0xDDD00001**, where DDD = dbspace number.

Multiple Extents – 1 Tblspace Tblspace

Chunk Pathname	Size	Used	Free
2 /chunks/94/dbs1	10000	103	9897

Description	Offset	Size
RESERVED PAGES	0	2
CHUNK FREELIST PAGE	2	1
dbs1:'informix'.TBLSpace	**3**	**50**
FREE	53	37
dbs1:'informix'.TBLSpace	**90**	**50**
FREE	140	9860

Total Used: 103
Total Free: 9897

Problem: Can't drop chunk

- Situation.
  1. Customer adds a chunk temporarily to a dbspace to create a large table in the dbspace.
  2. During processing the tblspace tblspace extends into the newly added chunk.
  3. The customer drops the large table and now wants to drop the chunk.

Problem: Can't drop chunk

IBM Informix Dynamic Server Version 9.40.UC2 -- On-Line -- Up 3 days 00:28:06 -- 29696 Kbytes

```
Dbspaces
address  number   flags      fchunk   nchunks  flags    owner    name
a0d0d7d8 1        0x20001    1        1        N        informix rootdbs
a152e4e8 2        0x21001    2        2        N        informix dbs1
 2 active, 2047 maximum

Chunks
address  chunk/dbs  offset     size       free       bpages     flags pathname
a0d0d928 1     1    0          40000      10630                 PO--  /spare2/chunks/rootchunk.940.0
a0e0be90 2     2    0          10000      3                     PO--  /chunks/94/dbs1
a153f6b0 3     2    0          10000      9947                  PO--  /chunks/94/dbs1.1
 3 active, 2047 maximum
```


Problem: Can't drop chunk

Chunk Pathname	Size	Used	Free
3 /chunks/94/dbs1.1	10000	53	9947

Description	Offset	Size
RESERVED PAGES	0	2
CHUNK FREELIST PAGE	2	1
FREE	3	32
dbs1:'informix'.TBLSpace	**35**	**50**
FREE	85	9915

Total Used: 53
Total Free: 9947

Resolution:

- Drop Dbspace
  - This causes the customer to have to unload the entire dbspace and drop all tables in the dbspace, then drop and recreate the dbspace.
  - This creates an outage
- Dialin To drop Chunk
  - Another possible solution was to have Advanced Support dial in and drop the chunk. They would also have to clean up the tblspace tblspace. This caused down time and is risky. (usually avoided)

Some Alternatives

- Truncation of tblspace tblspace
  - Only truncate if unused pages in last extents
  - Move partition page into earlier extents
- Moving of extents
  - oncheck –me
- For simplicity and flexibility the option was chosen to allow sizing of first and next extent sizes on the tblspace tblspace.

The feature

- Specify First/Next Extent sizes
  - When creating a dbspace you will now have the option to specify the first and next extent sizes.
  - When creating the instance (oninit –iy) you will be able to specify the first and next extent sizes of the root dbspace.
- New Algorithm for Extending Tblspace Tblspace
  - When creating an additional extent for a tblspace tblspace more emphasis is placed on the location of the extents.

First/Next Extent Sizes – Root dbspace

- Root dbspace (oninit –iy)
  - TBLTBLFIRST
    - This **onconfig parameter** is used to specify the size of the first extent of the tblspace tblspace for the root dbspace.
  - TBLTBLNEXT
    - This **onconfig parameter** is used to specify the next extent size of the tblspace tblspace for the root dbspace.
  - If these parameters are not present, the defaults will be used. Defaults will be discussed later. **All sizes are in KB**, and must be a multiple of the page size.

First/Next Extent Sizes – non root

- Non-Root dbspace (onspaces)
  - When creating a new dbspace with onspaces you will now be allowed to specify the first and next extent sizes for the tblspace tblspace.
    - -ef <first extent size> –en <next extent size>
  - If these options are not used the defaults will be used. Defaults will be discussed later. **All sizes are in KB**, and must be a multiple of the page size.

Example

```
onspaces –c –d dbs1 –p /spare2/dbs1.1 –o 0  \
        –s 10000 –ef 150 –en 150
```


Extending Tblspace Tblspace – non root

- Non-Root dbspace
  - When the tblspace tblspace is extending we now call tbltblalloc() instead of dbsalloc().
  - This new function has changed the way the tblspace tblspace creates new extents.

Extending Tblspace Tblspace – non root

Example

1. We have a dbspace “dbs1” with 2 chunks in it. Chunk #3, #4. We will assume the defaults of 50/50 for first/next extent sizes.
2. Chunk 3 has 40 contiguous free pages, and chunk 4 is empty.
3. dbsalloc() would extend and create a new extent in chunk 4.

Extending Tblspace Tblspace – non root

Example

4. tbltblalloc() will extend and create a new extent in chunk 3, with a size of 40. Even though we didn’t get the next size, instead we get the largest available size > the minimum extent size.
5. If the minimum extent size is not available then we would move to chunk #4 and try to get the next extent size.

Extending Tblspace Tblspace – non root

Search Order

1. 1st chunk in the dbspace for Next extent size
2. 1st chunk in the dbspace for next best size > min size.
3. 2nd chunk in the dbspace for Next extent size.
4. 2nd chunk in the dbspace for next best size > min size.
5. Next chunk in the dbspace for Next extent size.
6. Next chunk in the dbspace for next best size > min size.

Extending Tblspace Tblspace – root

Example

1. We create a root dbspace 3 chunks in it. Chunk #1, #2 and #3. We will use the defaults of 250/50 for first/next extent sizes.
2. Chunk 1 has 400 contiguous free pages, and chunk 2 has 40 contiguous free pages. Chunk 3 is empty.
3. dbsalloc() would extend and create a new extent in chunk 3, starting its search for free space in chunk #2, then moving to #3, then if not found, looking into chunk #1.

Extending Tblspace Tblspace – root

Example

4. tbltblalloc() will extend and create a new extent in chunk 2, with a size of 40. Even though we didn't get the next size, instead we get the largest available size > the minimum extent size.
5. We start the search in chunk 2, if the minimum extent size is not available then we would move to chunk #3 and try to get the next extent size. If that wasn't available we will get the next best size. Saving chunk #1 for last in our search for free space.

Extending Tblspace Tblspace – root

Search Order

1. 2nd chunk in the root dbspace for Next extent size.
2. 2nd chunk in the dbspace for next best size > min size.
3. Next chunk in the dbspace for Next extent size.
4. Next chunk in the dbspace for next best size > min size.
5. When all chunks have been searched then search the 1st chunk.

Reversion

- This feature will not prevent the reversion of any dbspaces.
- It will now be possible to revert back to a version prior to this feature, and now have extents for a tblspace tblspace that are not a default value or a doubling of that.
  - The next extent sizes will be adjust accordingly on the reversion process

Page 1

Administration : Multiple Table Fragments in Single Dbspace

- You can create partitions within a dbspace that can each support a table fragment.
  - reduces the total number of dbspaces needed for a fragmented table.
- Storing multiple table fragments in a single dbspace improves query performance over storing each fragmented expression in a different dbspace.
- This feature improves performance and simplifies management of dbspaces.

Page 1

Administration: Shared Memory > 4 GB

- You can now specify that segments for shared memory be created as large as:
  - your operating system platform, or
  - the SHMMAX parameter allows.

Page 1

Set Explain Enhancements

- SET EXPLAIN STATISTICS
- Onstat –g pqs {pid}
  - Print query statistics
- SQLSTATS = 1 or 2
  - 1: turn on counters
  - 2: add turning on timing

Page 1

Availability: Index Changes

- CREATE INDEX and DROP INDEX now supports DDL operations that apply no exclusive lock to the table on which the specified index is defined.
  - If you use this syntax to create an index on a table that other users are accessing, the index is not available until no user is updating the table.
  - After you issue the new syntax to drop an index, no one can reference the index, but concurrent DML operations can use the index until they terminate.
  - Dropping the index is deferred until no user is using the index.
  - This feature maintains the availability of the table within a production environment after an existing index has ceased to be efficient.

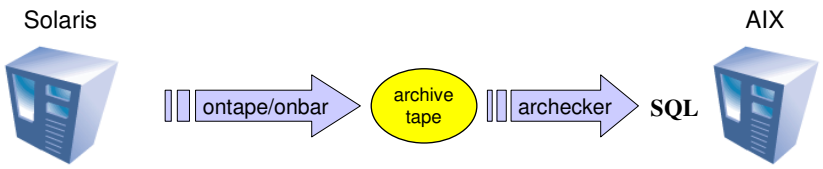
Limiting Memory Usage During ONLINE INDEX BUILD

- The **ONLIDX_MAXMEM** configuration parameter limits the amount of memory that is allocated to a single *preimage* pool and a single *updator* log pool.
  - The preimage and updator log pools, **pimage_<partnum>** and **ulog_<partnum>,** are shared memory pools that are created when a CREATE INDEX ONLINE statement is executed.
- Default: 5120 Kb. Minimum: 16Kb. Maximum: 4,294,967,295 kilobytes.
  - This configuration parameter in the ONCONFIG file before starting the database server, or you can change it dynamically through the **onmode -wf** or **onmode -wm** commands.

Page 1

Table Level Point-in-Time Restore (TLR)

- Purpose…to allow restoration of table data from a level 0 archive to a user specified point in time.
- Data may be restored to any *machine*.
- SQL Driven Distributed Restore (SDDR)
  - Extract a table or set of tables
  - You may apply a filter to the retrieved data
  - Retrieve just a subset of the columns
  - Repartitioning of the data



See IDS 10.0 Backup and Restore Guide (chapter 16)

Physical Restore

- Extracts requested data from a level 0 archive
- Two temporary tables create to store partial rows (i.e. rows not contained on a single page)

Physical Restore Flow

- The list of dbspaces is built from the create table statements in the schema command file
  - For onbar - only the dbspaces containing the table's data are scanned.
  - For ontape – currently all dbspaces scanned
- The data pages from the tables being restored are processed
  - Rows are extracted from the data pages
  - Converts to SQL Inserts or unloads format

Logical Recovery

- Replays the logical logs required to bring the data from the level 0 archive point in time(s) to the DBA requested restore point in time.
- Two parts to logical recovery
  - LOG STAGER
    - Reads the logical logs file(s)
    - Filters out unwanted records
    - Insert the required log records into log staging tables
  - LOG APPLIER
    - Reads the log staging tables
    - Applies the records to the destination tables

Schema Command File Example

```
database test1;
create table tlr    (
      a_serial   serial,
      b_char     char(20)
    ) in dbspace1;
create table tlr_dest    (
      a_serial   serial,
      b_char     char(20)
   ) in dbspace2;

insert into tlr_dest select * from tlr;

set workspace to dbs1,dbs2;

restore to '2003-01-01 01:01:01';
```

Restore command

- Point in time to which the data is restored
  - A quoted time
  - To restore to the most recent time use the keyword *CURRENT*
- To extract data only from a level 0 archive use the "*NO LOG RESTORE*"
- If you omit the entire statement the default is "*RESTORE TO CURRENT*"

COMMAND	REASON
RESTORE [TO "timestamp" | CURRENT] [NO LOG RESTORE]	**Set the time the restore should use and if logical logs should be used.** **NO LOG RESTORE causes the data to be extracted from the level 0 archive only**

Example Screen Output

```
archecker -bdvsX -f SETUP/setup_basic.cmd
AC_STORAGE          /tmp
AC_MSGPATH          /tmp/ac.log
AC_VERBOSE          on
AC_TAPEBLOCK          32 KB
AC_PAGESIZE         4096
AC_IXBAR            /vobs/tristarm/sqldist/etc/ixbar_olympia.85
Dropping old log control tables
Extracting table d1:t1 into d2:t1n
Extracting table d1:t2 into d2:t2n
Scan PASSED
Control page checks PASSED
Table checks PASSED
Table extraction commands 2
Tables/fragments found on archive 2
LOADED: d2:t2n produced 37 rows.
LOADED: d2:t1n produced 36 rows.
Creating log control tables
Staging Log 1:14
Staging Log 1:15
Logically recovered d2:t1n Inserted 602 Deleted 19 Updated 0
Logically recovered d2:t2n Inserted 0 Deleted 9 Updated 0
```

Tables being extracted

Rows extracted from Physical Restore

Logical Logs Processed

Logical Recovery Actions

Restoring a Table from a Previous Backup

- Extracts a table called *test1:tlr* from the level 0 backup of *dbspace1* just prior to "2003-01-01 01:01:01" and places the data in table *test1:tlr.*
- The logical logs will be applied up to "2003-01-01 01:01:01"

```
database test1;
create table tlr    (
       a_serial   serial,
       b_integer  integer,
       c_char     char(20),
       d_decimal  decimal,
       ) in dbspace1;
insert into tlr
  select * from tlr;
restore to
    '2003-01-01 01:01:01';
```


Distributed Restore

- Extracts a table called *test1:tlr_1* from the most recent backup of *dbspace1* and places the data on the database server *rem_srv* in the table *rem_dbs:tlr_1*

```
database rem_dbs;
create table tlr_1 (
   ( col1 int  )

database test1;
create table tlr_1
  (  col1 int  ) in dbspace1;

insert into rem_dbs@rem_srv.tlr_1
  select * from tlr_1;
```

Using External Tables

- Extracts a table called d1:source_tab from the most recent backup of dbspace1 and send the data in ASCII format with field delimited to the file named /tmp/pipe
- Could be useful with large tables in conjunction with HPL Express as the load utility
  - *Express uses light scans and no logging/constraint checking so it's typically very fast*

```
database d1;
create table source_tab
   ( cola int) in dbspace1;


create external table target_tab
  (  cola int  )
 USING ('/tmp/pipe', 'delimited');


insert into target_tab
  select * from source_tab;
```


acu_status

```
------------------------------------------------------------------------------
DBS:Table                 Status            Position       Stager    Worker
------------------------------------------------------------------------------
d1:t1                     Stage Complete    1:17:0          19051     19089
d1:t2                     Stage Complete    1:17:0          19051     19089

------------------------------------------------------------------------------
Number of Log records        1428     | Transaction count              18
TX Complete                     8     | TX with Begin only              0
Mismatch records                0     | TX with Commits                15
                                      | TX with Rollbacks               0
                                      | TX with Errors                  0
------------------------------------------------------------------------------
      Type                Count              Type                Count
------------------------------------------------------------------------------
  44  LG_RINSERT            688          40  LG_HINSERT            559
  41  LG_HDELETE              9
------------------------------------------------------------------------------
```

Data Types Not Support in the First Release

Extended Data Type
LIST
MULTISET
SET
ROW
DISTINCT
OPAQUE
Built-in Data Type

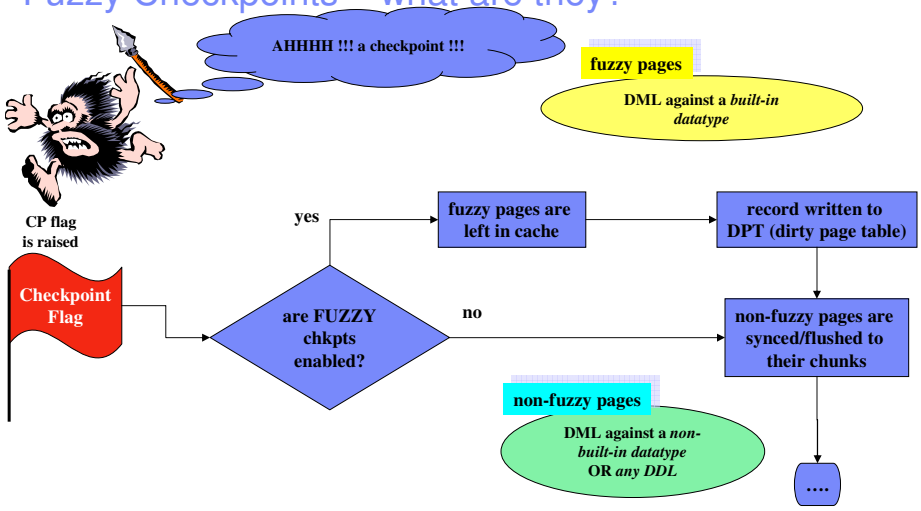
- This mean these data types can not be in any of tables being restored
- If these data types do appear an error will be given when processing the command file

Product Limitations

- The table schemas must be exact
- Since one of the goals of this feature is to recover accidentally dropped tables
  - Any drop of the fragment/table will be ignored during logical recovery
  - It will indicate the end of logical recovery for the table/fragment
- The tables/fragments being recovered must exist on the level 0 archive
- Stop logical recovery if an IPA has been performed since the level 0 restore
- A detached fragment is no longer part of the original table so recovery will be terminated on the detached fragment log record
- External tables are only physically restored
- Filters are only applied to tables which are physically restored

Page 1

Fuzzy Checkpoints – what are they?



Availability: Recovering Quickly with Fuzzy Checkpoints

- reduces the time required for engine recovery. This supports high availability by improving recovery performance when using fuzzy checkpoints.
- two new configuration parameters
  - FAST_RESTART_PHYSLOG – enables the flushing of DPT (dirty page table) records to the physical log on fuzzy checkpoints at checkpoint during the roll-forward phase of recovery.
  - FAST_RESTART_CKPT_FUZZYLOG – enables the engine to do physical logging on fuzzy checkpoints during the roll-forward phase of recovery.

Page 1

ER: Resending Indexes in HDR

- You can resend an index that became corrupt on the secondary server in an HDR pair.
- Resending an index is quicker than dropping and then rebuilding the index on the primary server.
- This feature increases the availability of the HDR primary server.

Page 1

ER Enhancements: Alter table/fragment support

- Overview
  - This feature provides alter support for tables being replicated via Enterprise Replication.
  - Currently, if the table schema needs to be altered or if the fragmentation strategy needs to be changed, then replication must be stopped, then alter is performed and then replication must be restarted. This is problematic as it makes it impossible to really consider ER in a 24X7 environment.

List of supported alter operations

1. The ability to add/drop default values
2. The ability to add/drop SQL checks
3. The ability to add/drop fragments
4. The ability to attach/detach fragments
5. The ability to add/drop columns
6. The ability to recluster indexes
7. The ability to alter non replicated columns

ER Enhancements: Alter table/fragment support

- Mastered Replicates
  - Currently no way to know if data types match between replicate nodes
  - Provides data type checking to eliminate possibility of corruption
  - New syscdr tables track data type information

ER Enhancements: Alter table/fragment support

- Remastering process
  - Existing replicate can be redefined by “remastering the replicate”. Through remastering process, a new column can be added/dropped to/from a replicate definition.
  - Also an existing non-mastered replicate can be converted to a mastered replicate using remastering process.

ER Enhancements: Alter table/fragment support

- Restrictions
  - ER must be in active state for altering a replicated table except in scenarios where adding/dropping check constraints and default values.
  - Alter operations are supported only on tables defined with mastered replicates.
  - Altering a replicated column size or data type is not supported. (Note: Modifying a non replicated column is supported).
  - Rename table operation is not supported
  - Rename column operation is not supported
  - Drop table operation is not supported

ER Enhancements: Replicate Templates

- Overview
  - Ease of ER administration and setup
    - The entire Enterprise Replication domain can be setup using simple commands with options such as defining replicates on all tables within database, specified on the command line or using a input file.
    - A template can perform an initial data synchronization on new servers being added to a template..
    - A template can optionally create tables during realization if they do not exist on target servers during template realization.
  - Eliminates most of the table Schema related errors
    - Templates use the master dictionary from the Master node to create these tables to ensure consistent schemas between the nodes.

ER: DRAUTO

- You can automate switching servers for High-Availability Data Replication if the primary server fails by using the DRAUTO configuration parameter.
- If DRAUTO is set to either
  - RETAIN_TYPE or REVERSE_TYPE, the secondary database server switches to type standard automatically when an HDR failure is detected.
    - OFF (0), The secondary periodically tries to reconnect to the primary
    - RETAIN_TYPE (1), the secondary switch to standard during a failure. When the HDR connection is restored, The original secondary database server shuts down and then switches back to type secondary.
    - REVERSE_TYPE (2), the original secondary database server switches to type primary as soon as the connection ends. When the HDR connection is restored, the original primary switches to type secondary.

Page 1

ER Enhancements: Replicate Resynch

Overview

- Meant to be used:
  - Bring a newly participating table up-to-date with the ongoing replication
  - Repair a replicated table if replication was stopped or failed for some reason.
- Two ways to repair a table:
  - Quick way: By processing the ats/ris files.
  - Other way: By defining and running a ‘resynch job’.

ER Enhancements: Replicate Resynch

Details of a resynch job:

- Creates a shadow replicate to carry the dummy updates as well as the forwarded data.
- Puts the replicate on target in 'skip' state – data coming from participants other than the source is skipped for the duration of the resynch.
- Puts the replicate on source in 'forward' state – data coming from other participants is forwarded to the target via 'shadow replicate'.
- Creates three internal tables which are replicated between source and target.
- Generates stored procedures and triggers to:
  - Scan the source data
  - Handle the extra target row options – including cascade deletes if the option is delete
  - Cleanup the rows in the internal tables and populate the violations table for errors.
  - Do dummy updates on the source data to replicate to the target.

ER Enhancements: Replicate Resynch

Listing the details of a repair job

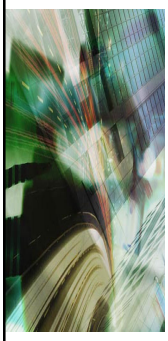
$ cdr list resynch

```
RESYNCHJOB          REPLICATE/REPLSET                STATE
-------------------------------------------------------------------------------
Repair_acct         Acct_repl                        Completed
Repair_txns         Txns_repl                        Defined
```

$ cdr list resynch Repair_acct

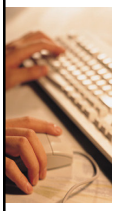
```
RESYNCHJOB          REPLICATE/REPLSET                STATE
-------------------------------------------------------------------------------
Repair_acct         Acct_repl                        Completed

SOURCE
------
test@g_serv1:informix.account
      select * from 'informix'.account

TARGET
------
test@g_serv2:informix.account
      select col1,col2 from 'informix'.account

BLOCK SIZE:                 10
TARGET ROW OPTION:          Delete
PROCESSED ROWS:             100
START TIME:                 2004-05-25 13:58:36
END TIME:                   2004-05-25 15:34:36
```

IDS v10.0 Backup and Restore Enhancements



- View logical logs backed up by OnBAR
  - Similar to using OnLog utility
- OnBAR debugging level can be changed while OnBar is running
  - Saves time and disk space
  - Bar_Debug can be set as frequently as needed
- Tivoli Storage Manager XBSA is included with IDS

IDS v10.0 Application Development



- Enterprise Generation Language (EGL) is a core technology in:
  - Rational Web Developer
  - Rational Application Developer
  - Rational Software Architect
  - WebSphere Application Server – Express
- EGL is ideal for Data-driven Rapid Application Development:
  - Web, Text, and Batch Applications
  - With IBM Databases (IDS v10.x and DB2 8.x)
- EGL generates Java at runtime
- Majority of Informix 4GL language constructs are included in the EGL
- Informix 4GL to EGL Conversion Utility is included with V6.0 products
  - GA with IDS v10, Q1 2005

Applications: ESQL/C to DB2

- You can run Informix ESQL/C applications with DB2 servers and databases.
- The Informix ESQL/C product provides a new library that is called when you use the **esql** command to preprocess your files to work with DB2.
- Informix ESQL/C runs with DB2 Version 8.2, or later, running on Linux, UNIX, and Windows operating systems.

Page 1

Applications: JDBC 3.0 Support

- Version 3.0 of the IBM Informix JDBC Driver supports the following features in compliance with the Sun Microsystems JDBC 3.0 specification:
  - Internally update BLOB and CLOB data types using all methods introduced in the JDBC 3.0 specification.
  - Specify and control ResultSet holdability, leveraging on the Informix JDBC extension implementation.
  - Retrieve auto-generated keys from the database server.
  - Access multiple INOUT mode parameters in Dynamic Server through the CallableStatement interface.
  - Provide a valid large object descriptor and data to the JDBC client to send or retrieve BINARY data types as OUT parameters.
  - J/Foundation supports JRE Version 1.4 and the JDBC 3.0 specification.

Page 1

Applications: Full .NET Support

- The .NET Provider enables Windows .NET applications to access and manipulate data in IBM Informix databases.
  - The IBM Informix .NET Provider is a runtime library that encapsulates a data access API for use by Microsoft .NET applications.
  - It consists of a set of specialized classes that implement standard Microsoft ADO.NET interfaces and serves as a bridge between IBM Informix databases (data sources) and .NET applications.
- Windows client applications written in any .NET supported language can take advantage of the IBM Informix .NET Provider. Some examples of client applications are:
  - Visual BASIC .NET applications
  - Visual C# .NET applications
  - Visual J# .NET applications
  - ASP.NET web applications
- The IBM Informix .NET Provider a connection editor dialog box, a command editor dialog box, and a data adapter wizard as Microsoft Visual Studio add-ins.

Page 1

Update: Fixpack 3 (July 2005)

- Transaction support for XA-compliant external data sources
- MQ DataBlade module
- New DBCREATE_PERMISSION configuration parameter to restrict the ability to create databases
- New secure default directory for the DUMPDIR configuration parameter
- Table-level restore for smart large object columns
- AES cipher support for network encryption
- New Enterprise Replication commands to show statistics information
- Client SDK included in Dynamic Server installation process
- Returning subsets of query results as collection-derived tables
- Ordering subsets of query results in collection-derived tables
- J/Foundation upgrade to JRE 1.4.2
- New default directory for ADTPATH configuration parameter
- New UNSECURE_ONSTAT configuration parameter

Page 1

Update: Fixpack 4 (December 2005)

- TRUNCATE TABLE
- Enterprise Replication direct synchronization
- Enterprise Replication consistency checking
- Enhanced support for IPv6
- Secure local connections
  - SECURITY_LOCALCONNECTION onconfig parameter
- Secure DataBlade module paths
  - DB_LIBRARY_PATH onconfig parameter
- Parallel backup and restore is more efficient
  - BAR_SORT_DBS environment variable
- DB-Access stops processing after the first error
- Informix Interface for TSM supports HP-UX (Itanium)
- New default value for IFX_EXTEND_ROLE

Page 1

End
IDS 10.0
Technical Deep Dive

Helpful Links & Contacts

- Informix Product Page - http://www.ibm.com/informix/
- Informix Connection Program (ICON) - http://www.informixconnect.com/icon/
- Informix Platform Roadmap - http://www-306.ibm.com/software/data/informix/pubs/roadmaps.html
- Informix Product Life Cycle (PLC) Roadmap - http://www-306.ibm.com/software/data/informix/support/plc/
- Informix International User Group (IIUG) - http://www.iiug.org
- DM Today Newsletter - http://www-3.ibm.com/software/data/db2infonews/