IBM Informix v.14.10.xC6 Technical Deep Dive Webcast series

Session 1: Kernel, Enterprise Replication, and SQL features

Session 2, July 13 - IHQ and backup from RS secondaries

Carlton Doe
World Wide Informix team
cdoe@us.ibm.com

IIUG Virtual event
October 5 - 7, 2021

- Three half days of hour-log technical sessions, keynotes, and tutorials
- NEW v.14.10 badge exam will be available as part of your paid registration
- Registration information coming soon
  - Watch for more information on iiug.org, the *IIUG Insider*, IIUG email and the IBM Community site

IBM Informix v.14.10.xC6 Technical Deep Dive Webcast series

Session 1: Kernel, Enterprise Replication, and SQL features

Session 2, July 13 - IHQ and backup from RS secondaries

Carlton Doe
World Wide Informix team
cdoe@us.ibm.com

IBM INFORMIX V.14.10.XC6 - MOVING TO XC6

v.1a

Conversion requirement

- In Informix 12.10.xC5, rolling upgrades of H/A clusters was introduced
  - Rolling upgrades enable intra-version upgrades
    - From one fixpak to the next fixpak
      - For example, xC4 to xC5 or xC5 to xC6
      - Can NOT go from xC4 to xC6
  - Within a fixpak, from one interim to the next interim or from an interim to the next fixpak
    - For example, xC5 to xC5W1 or xC5W3 to xC5W4 or xC4W2 to xC5
- We said there are two internal controls which prevent moving from xC4 to xC6 for example
- We also there may be cases where we prevent rolling upgrades even between normally allowed upgrades
  - Disk structures are changed
  - Existing logical log records appear to be modified
  - The new version requires conversion or system catalog changes as part of first-boot of the new software version

This is occurring in xC6!

Conversion requirement

- For xC6, if you are using an H/A cluster, you *will need to take an outage* to migrate from an earlier version
  - Turn off the cluster
  - Load xC6 binary on each node, update the `$ONCONFIG` files
  - Turn on the primary, let it convert, leave it online
  - One at a time, turn on each secondary, let it convert and connect to the primary
- As you will see in a moment, ER requires a conversion as well

IBM INFORMIX V.14.10.XC6 - KERNEL ENHANCEMENTS

v.1b

Agenda

- pSeries memory accessibility / performance fix
- Query runtime and automatic timeout
- Last DML in a table
- Ability to use temp dbspaces of differing page sizes
- Parallelized SLOBspace cleaning
- `ifx_row_id` access enhancements
- Automatic semi-`AUTOLOCATE` functionality with round robin fragmentation
- TLS 1.3 support

Some screenshots courtesy of JC Lengyel

pSeries memory accessibility / performance fix

pSeries memory access

- With the IBM p8 and especially p9 series running AIX, customers experienced “challenges” getting optimal performance from the machines
  - It became worse as the workload increased
  - The root cause of the performance degradation has been identified and changes to the AIX port have been made
- Historically, Informix shared memory address blocks have begun on 256 MB address boundaries

```
$ onstat -g seg

IBM Informix Dynamic Server Version 14.10.FC6 -- On-Line -- Up 00:21:27 -- 411648 Kbytes
2021-04-07 08:04:27

Segment Summary:
id         key        addr             size             ovhd     class blkused  blkfree
59769980   528c4801   700000010000000  4947968          499288   R     1204     4
65012859   528c4802   700000020000000  33439744         393384   V     6900     1264
144704654  528c4803   700000030000000  216104960        1        B     52760    0
60818554   528c4804   700000040000000  166461440        1        B     40640    0
119538826  528c4805   700000050000000  573440           7992     M     136      4
Total:     -          -                421527552        -        -     101640   1272
```

pSeries memory access

- On p9 hardware using AIX, it was discovered that the kernel is using a memory address translation table with 12 slots that are aligned on a 1 TB memory address boundary
  - With Informix's 256 MB address scheme more and more memory location address "misses" occurred as instance workload and memory utilization increased resulting in poor performance

- For AIX ports, a new configuration parameter is available with the default set to "on"

  `AIX_LSA`

  `0` (zero) — off, use legacy 256 MB address boundaries

  `1` (one) — on, default, align to 1 TB memory address boundaries

  `2` (two) — align to 1 TB boundary and allow resident and first virtual portion to share an address

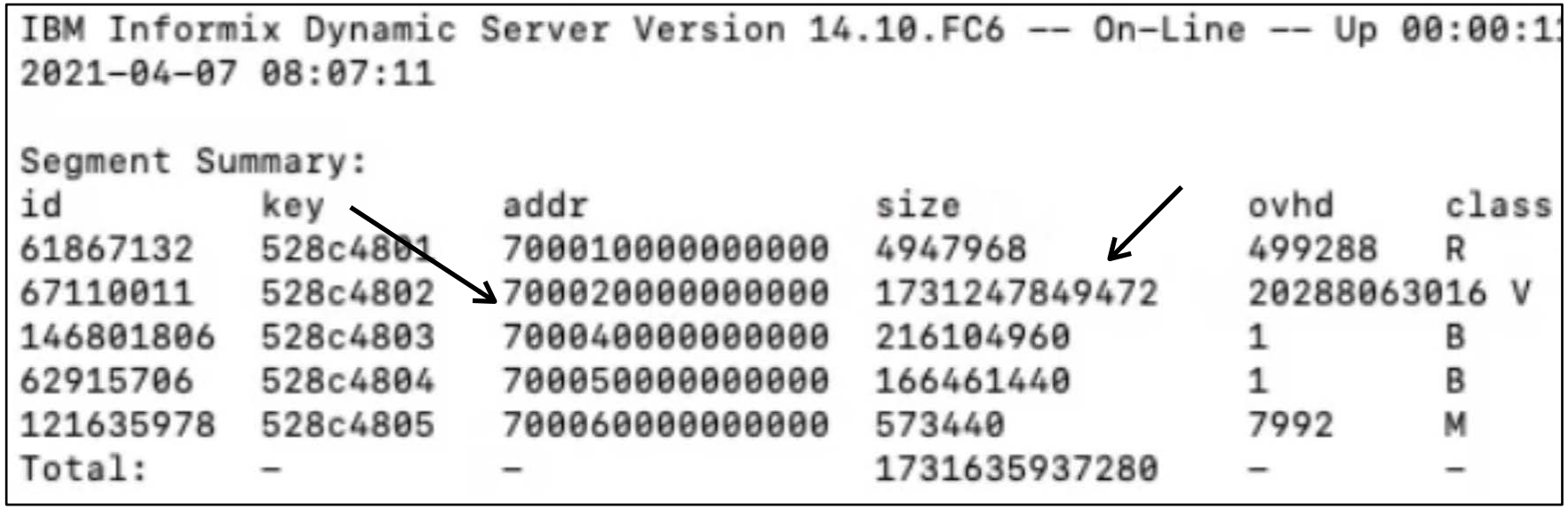
```
IBM Informix Dynamic Server Version 14.10.FC6 -- On-Line -- Up 00:00:32 -- 411648 Kbytes
2021-04-07 08:06:28

Segment Summary:
id          key         addr              size        ovhd     class blkused  blkfree
60818556    528c4801    700010000000000   4947968     499288   R     1204     4
120587402   528c4802    700020000000000   33439744    393384   V     6933     1231
61867130    528c4803    700030000000000   216104960   1        B     52760    0
145753230   528c4804    700040000000000   166461440   1        B     40640    0
66061435    528c4805    700050000000000   573440      7992     M     136      4
Total:      -           -                 421527552   -        -     101673   1239
```

pSeries memory access

- This does NOT affect the size of the shared memory segments
  - They are NOT required to be 1 TB in size
  - It also does not affect how many shared memory segments exist in the instance
    - A caveat to this in just a moment
  - What is happening under the covers is the engine is (where necessary) changing `shmbase` to a new address with a different offset automatically to match the O/S
    - If memory segment exceeds 1 TB, the next segment will automatically jump to the next available address location

For example, a 1.7 TB virtual segment forces the first buffer pool segment to skip to the next available address to conform to the boundary



```
IBM Informix Dynamic Server Version 14.10.FC6 -- On-Line -- Up 00:00:11
2021-04-07 08:07:11

Segment Summary:
id          key       addr              size           ovhd       class
61867132    528c4801  700010000000000   4947968        499288     R
67110011    528c4802  700020000000000   1731247849472  20288063016 V
146801806   528c4803  700040000000000   216104960      1          B
62915706    528c4804  700050000000000   166461440      1          B
121635978   528c4805  700060000000000   573440         7992       M
Total:      -         -                 1731635937280  -          -
```

pSeries memory access

- As mentioned earlier, the AIX memory address translation table has 12 slots
  - If you exceed 12, performance will start to degrade
    - It's not a dramatic decline on the 13th segment but you will see it
  - On Solaris it is possible for the resident and first virtual portion of shared memory to share the same memory segment address
    - This is now possible on AIX to reduce the number of segments
      - Set `AIX_LSA` to 2

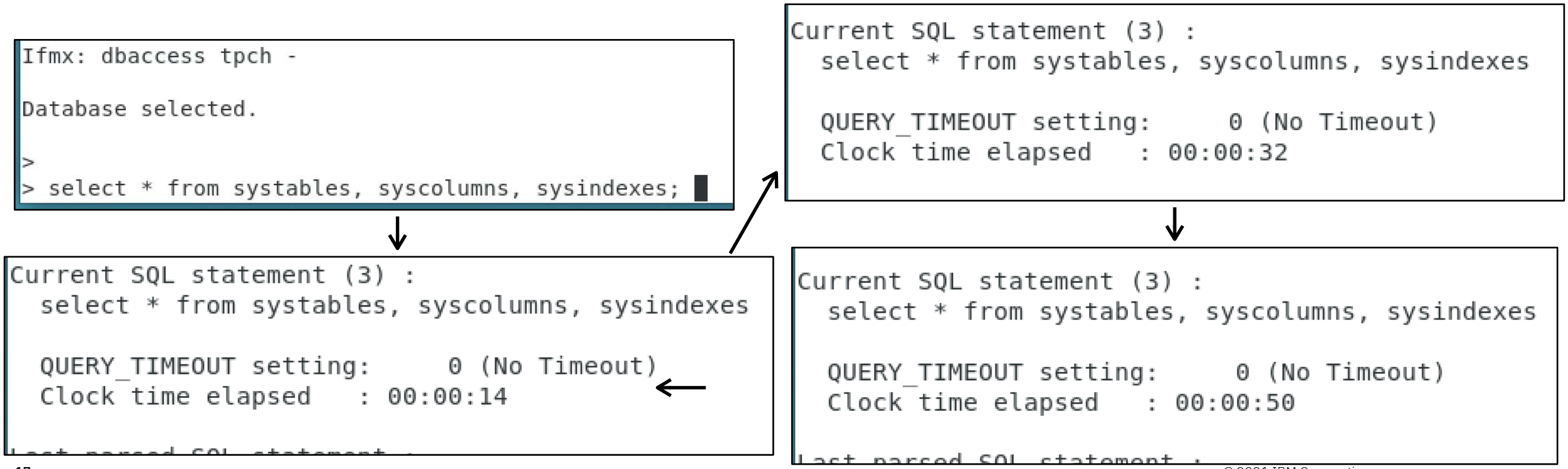
```
IBM Informix Dynamic Server Version 14.10.FC6 -- On-Line -- Up 00:00:12
2021-04-07 08:08:17

Segment Summary:
id          key         addr               size               ovhd      class
62915708    528c4801    700010000000000    20774141952        24388885  R
 (shared)   528c4801    7000104d63c4000    1751535927296      205258128 V
122684554   528c4802    700030000000000    216104960          1         B
```

Query runtime information / automatic query timeout

Query runtime / timeout

- A query's runtime is now available
  - The `onstat -g [ ses | sql ] sess_id` output now includes information about the query timeout value and query runtime for a *running* query
  - This query is fairly simple, but takes time to complete
    - It was interrupted before completion



Query runtime / timeout

- Queries that complete immediately or are completed/ended, do not report time values
  - It looks like pre-xC6 output

```
Sess       SQL             Current                  Iso Lock
Id         Stmt type       Database                 Lvl Mode
4          -               tpch                     CR  Not Wa

Last parsed SQL statement :
  select * from customer where c_custkey = 100

Ifmx:
```

Query runtime / timeout

- Customers have looked for ways to tame out of control queries for a while
  - “Hey, let’s do a left outer join across (7) 1 billion row tables!!!!”
- Informix v.14.10.xC6 introduces a new session-level environment variable to automatically interrupt long running queries

  `QUERY_TIMEOUT` *n*

  where n is between `0` (zero) (off) to `10000` seconds
  - The variable ***is*** case sensitive, lower case won’t work
  - This variable only affects queries, not DML operations
    - You don’t need to worry about batch loads / updates failing
  - It must be set for each user session, not once when the instance is started for everyone
- Currently, if a session’s query is interrupted, no message is logged to `MSGPATH` but that will change in the future
- Also, the application of the timeout may not be exactly N seconds
  - It could be slightly longer depending on where the query is with respect to internal break points when the runtime can compared to the timeout value

Query runtime / timeout

- Set the `QUERY_TIMEOUT` environment variable and try the operation again

```
Ifmx: export QUERY_TIMEOUT=20
Ifmx:
Ifmx: env|grep QUERY
QUERY_TIMEOUT=20
```

```
Ifmx: dbaccess tpch -

Database selected.

>
> select * from systables, syscolumns, sysindexes;
```

Or

This approach gives you flexibility to enable, change the timeout value, or disable it completely, throughout application processing as needed

```
Ifmx: dbaccess tpch -

Database selected.

>
> set environment QUERY_TIMEOUT '20';

Environment set.

>
> select * from systables, syscolumns, sysindexes;
```

Query runtime / timeout

- The query runs until the timeout value is reached

```
Current SQL statement (3) :
  select * from systables, syscolumns, sysindexes

  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:06
```

```
Current SQL statement (3) :
  select * from systables, syscolumns, sysin

  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:15
```

Query is timed out →

```
  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:18

Last parsed SQL statement :
  select * from systables, syscolumns, sysindexes

Ifmx: onstat -g sql 67

IBM Informix Dynamic Server Version 14.10.FC6DE -- On-Line -
2021-05-14 16:03:15


Sess        SQL             Current            Iso Lock
Id          Stmt type       Database           Lvl Mode
67          -               tpch               CR  Not Wait

Last parsed SQL statement :
  select * from systables, syscolumns, sysindexes

Ifmx:
```

Query runtime / timeout

- On the client side, the session receives a `-213` error
  - Should be added to the application `whenever error` functionality

```
clust             14.00000000000

  213: Statement timed out or interrupted by user.

  157: ISAM error: Interrupted ISAM call
Error in line 1
Near character position 46
>
```

Query runtime / timeout

- There is an interesting nuance to this feature
  - If you use `dbaccess` in ***menu*** mode, the timeout value doesn't really work if you pause on a result screen
  - Even using the `Next` option doesn't force the timeout to occur
    - This may be due to the utility caching results
      - Once the cache is used and it asked for more results from the instance, the operation may time out

```
----------------------- tpch@inst_1 -----


c_custkey     3
c_name        Customer#000000003
c_address     MG9kdTD2WBHm
c_nationkey   1
c_phone       11-719-748-336
c_acctbal     7498.120000000
c_mktsegment  AUTOMOBIL
c_comment     special packages wake. slyl
```

```
Current SQL statement (2) :
  select * from customer

  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:09
```

```
Current SQL statement (2) :
  select * from customer

  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:18

Last parsed SQL statement :
```

```
Current SQL statement (2) :
  select * from customer

  QUERY_TIMEOUT setting: 00:00:20
  Clock time elapsed   : 00:00:29
```

Last DML in a table

Last table DML time

- `oncheck -pt` *`tabname`* shows the last time a table was updated
  - This works for fragmented as well as unfragmented tables
  - It's built on top of the .xC2 functionality describing when an index was last used used

  - In this example, a simple table is built then rows are added with `col1` values `1` and `2`

```
create table my_tab (
 col1 int,
 col2 varchar(12)
)
fragment by expression
 col1 = 1 in data_space_1,
 col1 = 2 in data_space_2,
 remainder in data_space_3;
```

```
----------------------- tpch@inst_1 ----

insert into my_tab values (1, "text");
insert into my_tab values (2, "text");
insert into my_tab values (1, "text");
insert into my_tab values (2, "text");
insert into my_tab values (1, "text");
insert into my_tab values (2, "text");
insert into my_tab values (1, "text");
insert into my_tab values (2, "text");
```

Last table DML time

- Executing the `oncheck` command shows updates to the space 1 and 2 but not the remainder space

```
             Table fragment partition data_space_1 in D

Physical Address                2:638
Creation date                   05/14/2021 16:44:33
TBLspace Flags                  902        Row Locking
                                           TBLspace conta
                                           TBLspace use 4

Maximum row size                17
Number of special columns       1
Number of keys                  0
Number of extents               1
Current serial value            1
Current SERIAL8 value           1
Current BIGSERIAL value         1
Current REFID value             1
Pagesize (k)                    2
First extent size               8
Next extent size                8
Number of pages allocated       8
Number of pages used            2
Number of data pages            1
Number of rows                  4
Partition partnum               2097291
Partition lockid                2097291
Last DML time                   Fri May 14 16:46:16 2021
```

```
     Table fragment partition data_space_2 i

Address                 3:5
date                    05/14/2021 16:44:33
Flags                   902        Row Locking
                                   TBLspace co
                                   TBLspace us

ow size                 17
 special columns        1
 keys                   0
 extents                1
erial value             1
ERIAL8 value            1
IGSERIAL value          1
EFID value              1
(k)                     2
ent size                8
nt size                 8
 pages allocated        8
 pages used             2
Number of data pages    1
Number of rows          4
Partition partnum       3145730
Partition lockid        2097291
Last DML time           Fri May 14 16:46:16 2021
```

```
             Table fragment partition data_space_3 i

Physical Address                4:5
Creation date                   05/14/2021 16:44:33
TBLspace Flags                  902        Row Locking
                                           TBLspace co
                                           TBLspace us

Maximum row size                17
Number of special columns       1
Number of keys                  0
Number of extents               0
Current serial value            1
Current SERIAL8 value           1
Current BIGSERIAL value         1
Current REFID value             1
Pagesize (k)                    2
First extent size               8
Next extent size                8
Number of pages allocated       0
Number of pages used            0
Number of data pages            0
Number of rows                  0
Partition partnum               4194306
Partition lockid                2097291
Last DML time                   No Record
```

Last table DML time

- Adding an additional row to the first fragment updates its DML timestamp but not the other fragments

```
Ifmx: oncheck -pt tpch:my_tab |grep DML
    Last DML time                   Fri May 14 17:03:10 2021
    Last DML time                   Fri May 14 16:46:16 2021
    Last DML time                   No Record
Ifmx:
```

Last table DML time

- If you add an index to the table, its fragment and access time is included

Index creation

```
            Index my_ind fragment partition data_space_5

Physical Address               6:5
Creation date                  05/14/2021 17:06:13
TBLspace Flags                 802        Row Locking
                                          TBLspace use 4 bi
Maximum row size               17
Number of special columns      0
Number of keys                 1
Number of extents              1
Current serial value           1
Current SERIAL8 value          1
Current BIGSERIAL value        1
Current REFID value            1
Pagesize (k)                   2
First extent size              4
Next extent size               4
Number of pages allocated      4
Number of pages used           3
Number of data pages           0
Number of rows                 0
Partition partnum              6291458
Partition lockid               2097291
Last Lookup/Scan               No Record
```

Statistics updated and indexed query executed

```
            Index my_ind fragment partition data_space_

Physical Address               6:5
Creation date                  05/14/2021 17:06:13
TBLspace Flags                 802        Row Locking
                                          TBLspace use 4
Maximum row size               17
Number of special columns      0
Number of keys                 1
Number of extents              1
Current serial value           1
Current SERIAL8 value          1
Current BIGSERIAL value        1
Current REFID value            1
Pagesize (k)                   2
First extent size              4
Next extent size               4
Number of pages allocated      4
Number of pages used           3
Number of data pages           0
Number of rows                 0
Partition partnum              6291458
Partition lockid               2097291
Last Lookup/Scan               Fri May 14 17:12:14 2021
```

Temp dbspaces of different pagesize

Temp dbspaces of differing pagesize

- The ability to create spaces of different page sizes has been in Informix for quite a while
  - They can be used to reduce wasted space and improve I/O efficiency

- Temporary spaces can be created in different sizes as well
  - For example in Informix v.14.xC3

```
Ifmx: onspaces -c -d work_space_4k -t -k 4 -p /opt/IB
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
Ifmx: onspaces -c -d work_space_12k -t -k 12 -p /opt/
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
```

```
Ifmx: onspaces -c -d work_space_16k -t -k 16 -p /opt/
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
```

```
2048     N TB     informix work_space
2048     N SB     informix slob_space
4096     N TB     informix work_space_4k
12288    N TB     informix work_space_12k
16384    N TB     informix work_space_16k
```

Temp dbspaces of differing pagesize

- Just because they can exist with different sizes, doesn't mean they can be used
  - Again in .xC3

```
Ifmx: onstat -g cfg dbspacetemp

IBM Informix Dynamic Server Version 14.10.FC3DE

name                         current value
DBSPACETEMP                  work_space
```

```
Ifmx: onmode -wf DBSPACETEMP=work_space,work_space_4k,work_space_12k,work_space_16k
Value for DBSPACETEMP (work_space) was saved in config file.
Value of DBSPACETEMP has been changed to work_space.
Ifmx:
Ifmx: onstat -g cfg dbspacetemp

IBM Informix Dynamic Server Version 14.10.FC3DE -- On-Line -- Up 02:02:00 -- 729796 Kbytes

name                         current value
DBSPACETEMP                  work_space

Ifmx:
Ifmx:
```

Temp dbspaces of differing pagesize

- Prior to xC6, all temp dbspaces had to be the same size
  - All 2k, 4k etc.

- We strongly recommended they be left at the instance default
  - So restores (particularly warm) could successfully execute
    - Logical log records requiring temporary storage during the restore can only be written to a temp space of default page size

- With xC6 you can have and successfully configure temp spaces of differing page sizes
  - We still strongly recommend at least one use the default page size

Temp dbspaces of differing pagesize

- For example

```
Ifmx: onstat -g cfg dbspacetemp

IBM Informix Dynamic Server Version 14.10.FC6DE -
2021-05-14 19:37:30

name                         current value
DBSPACETEMP                  work_space

Ifmx:
```

```
2048      N TB      informix work_space
2048      N SB      informix slob_space
4096      N TB      informix work_space_4k
12288     N TB      informix work_space_12k
16384     N TB      informix work_space_16k
```

```
Ifmx: onspaces -c -d work_space_4k -t -k 4 -p /opt/IBM/inf
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
Ifmx: onspaces -c -d work_space_12k -t -k 12 -p /opt/IBM/i
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
Ifmx: onspaces -c -d work_space_16k -t -k 16 -p /opt/IBM/i
Verifying physical disk space, please wait ...
Space successfully added.
Ifmx:
```

```
Ifmx: onmode -wf DBSPACETEMP=work_space,work_space_4k,work_space_12k,work_space_16k
Value for DBSPACETEMP (work_space,work_space_4k,work_space_12k,work_space_16k) was saved in config file.
Value of DBSPACETEMP has been changed to work_space,work_space_4k,work_space_12k,work_space_16k.
Ifmx:
Ifmx: onstat -g cfg dbspacetemp

IBM Informix Dynamic Server Version 14.10.FC6DE -- On-Line -- Up 00:06:21 -- 730464 Kbytes
2021-05-14 19:38:55

name                         current value
DBSPACETEMP                  work_space,work_space_4k,work_space_12k,work_space_16k

Ifmx:
```

Temp dbspaces of differing pagesize

- Great, I can have and use different page sizes, what does it mean to me??
  - When you create explicit temp tables, you can put them in the space whose page size most closely matches the row length so you don’t have home and reminder row splits
  - Without an explicit placement clause, the instance will create the table in the least used space without regard to row length

Parallelized SLOBspace cleaning

Parallelized SLOBspace cleaning

- Garbage collection within SLOBspaces occurs when the instance is started
  - Prior to xC6, it was a serialized operation
  - Normally, there isn't much that needs to be done so it doesn't take too long
  - However if there is an abnormal shutdown and a significant amount of SLOB DML operations were occurring more recovery is required.
- With xC6, the `SBSPACE_CLEANERS` parameter controls the number of parallelized threads for garbage collection on startup
  - With a range of 1 to 2,047, you can control the number of cleaner threads
    - If you set it to fewer than the number of SLOBspaces, it will work on that number in parallel
      - As one finishes, the thread will start on another space
    - If you set it greater than the number of SLOBspaces the instance will only fork enough for the available SLOBspaces
  - This parameter is *not* dynamically tunable

`ifx_row_id` access enhancements

ifx_row_id access enhancements

- Whether a table is explicitly created `with rowids` or not, the `ifx_row_id` pseudo-column exists and can be used
  - For example:

```
> select ifx_row_id from region;


ifx_row_id

7340100:257
7340100:258
7340100:259
7340100:260
7340100:261

5 row(s) retrieved.
```

  - The format of the `ifx_row_id` is a `varchar` with two values:
    - `fragment_id`
    - `row_id`

ifx_row_id access enhancements

- Using `ifx_row_id` as a predicate is possible but prior to xC6, it involved a sequential scan of the table
  - Slow!!!
  - For example on xC4:

```
Ifmx: more line_q.sql
select * from lineitem where ifx_row_id = "7340106:6800391";
Ifmx:
```

```
Ifmx: time dbaccess tpch line_q.sql

Database selected.



l_orderkey       400070
l_partkey        23074
l_suppkey        1843
l_linenumber     1
l_quantity       31
l_extendedprice  30909.17000000
l_discount       0.02
l_tax            0.06
l_returnflag     N
l_linestatus     O
l_shipdate       12/16/1995
l_commitdate     01/18/1996
l_receiptdate    12/29/1995
l_shipinstruct   DELIVER IN PERSON
l_shipmode       REG AIR
l_comment        slyly regular requests use. qui

1 row(s) retrieved.


Database closed.


real    0m2.048s
user    0m0.004s
sys     0m0.007s
Ifmx:
```

ifx_row_id access enhancements

- With xC6 *and exact matches*, the performance is significantly better
  - If ranges or other “soft” values are used, it’s still a sequential scan because of the `varchar` data type
    - Due to collation within the character string

xC4 time

```
Database closed.


real    0m2.048s
user    0m0.004s
sys     0m0.007s
Ifmx:
```

```
Ifmx: time dbaccess tpch line_q.sql

Database selected.



l_orderkey       400070
l_partkey        23074
l_suppkey        1843
l_linenumber     1
l_quantity       31
l_extendedprice  30909.17000000
l_discount       0.02
l_tax            0.06
l_returnflag     N
l_linestatus     O
l_shipdate       12/16/1995
l_commitdate     01/18/1996
l_receiptdate    12/29/1995
l_shipinstruct   DELIVER IN PERSON
l_shipmode       REG AIR
l_comment        slyly regular requests use. qui

1 row(s) retrieved.


Database closed.


real    0m0.440s
user    0m0.007s
sys     0m0.011s
Ifmx:
```

Semi `autolocate` with round robin fragmentation

Round robin fragmentation enhancement

- When a table is created with round robin fragmentation, rows are evenly inserted between all table fragments
  - Assuming there are no deletes, the fragments will grow in size uniformly over time
  - For example this small table with 5,000 rows in two fragments

```
Ifmx: oncheck -pt tpch:my_lineitem |grep "Number of rows"
    Number of rows                  2500
    Number of rows                  2500
Ifmx:
```

Round robin fragmentation enhancement

- If an additional fragment is added, it receives rows one-at-a-time like the other fragments
  - For example, an additional fragment is added then 3 rows are inserted

Before rows added

```
Ifmx: oncheck -pt tpch:my_lineitem |grep "Number of rows"
    Number of rows                  2500
    Number of rows                  2500
    Number of rows                  0
Ifmx:
```

After rows added

```
Ifmx: oncheck -pt tpch:my_lineitem |grep "Number of rows"
    Number of rows                  2501
    Number of rows                  2501
    Number of rows                  1
Ifmx:
```

Round robin fragmentation enhancement

- Informix v.14.xC6 adds functionality like `autolocate` to round robin fragmented tables
  - When new rows are inserted, they are put in the fragment with the least number of rows
  - This continues until the fragments are reasonably balanced then the inserts are distributed across all fragments again

- The phrase “reasonably balanced” is important
  - The row count check happens every 100 rows
    - It’s likely that “evenly balanced” table fragments will be off by up to 100 rows between the fragments
    - For example, in a three fragment table:
      - Fragment 1 = 300 rows
      - Fragment 2 = 300 rows
      - Fragment 3 = between 200 and 300 rows

Round robin fragmentation enhancement

- With the functionality enabled, the two fragment table is recreated and 5000 rows inserted

```
Ifmx: oncheck -pt tpch:my_lineitem |grep "Number of rows"
    Number of rows                  2500
    Number of rows                  2500
Ifmx:
```

- Another fragment is added and 70 rows inserted
  - They are all inserted into the new fragment
    - This continues until the new fragment has the same row count as the other fragments

```
Ifmx: oncheck -pt tpch:my_lineitem |grep "Number of rows"
    Number of rows                  2500
    Number of rows                  2500
    Number of rows                  70
Ifmx:
```

Round robin fragmentation enhancement

- Control of this functionality is not formally documented
  - A new, dynamically tunable `onconfig` parameter controls it
  - `LEGACY_RR` *`bool_val`*
    - `0` (zero) — false, default, row inserts favor the least used fragment
    - `1` (one) — true, legacy one-by-one distribution occurs

```
Ifmx: onstat -g cfg full LEGACY_RR

IBM Informix Dynamic Server Version 14.10.FC6DE -- On-Line -- Up 00:55:59
2021-05-14 21:57:28

Configuration Parameter Info

id   name                      type    maxlen   units   rsvd  tunable
257  LEGACY_RR                 BOOL    2                      *

     default : 0
     onconfig: 0
     current : 0

     This parameter is undocumented.

     Description:
     Enable LEGACY_RR to revert to the original round-robining algorithm,
     which does not take into account the number of rows in a fragment.
     This legacy algorithm is not recommended when AUTOLOCATE is enabled.
```

Round robin fragmentation enhancement

- As mentioned in the parameter documentation on the previous slide, if `AUTOLOCATE` is enabled, disable this parameter since `AUTOLOCATE` performs a much broader set of data placement algorithms and controls

TLS 1.3 support

TLS 1.3 support

- With xC6, Informix now supports TLS 1.3 for network connections with all its enhancements and increased security
  - This is the successor to SSL
- The default configuration is TLS 1.2
  - You use the `TLS_VERSION` configuration parameter to specify the version used
- There is a caveat to this functionality however
  - IBM GSKit TLS 1.3 does ***not*** support FIPS (Federal Information Processing Standard) mode yet
    - If you are using GSKit and specify FIPS, the instance will default to TLS 1.2
    - As a result, we strongly recommend disabling FIPS for GSKit TLS clients
      - Use the `ISI_GSK_FIPS_MODE=off` environment variable
    - ODBC and other clients are *not* affected by this FIPS / TLS 1.3 issue, just GSKit clients

Questions

IBM INFORMIX V.14.10.XC6 - SQL FEATURES

v.1a

Agenda

- Forwarding audit records to `syslog()`
- SQL interface to temp space usage
- Let accessed information for stored procedures
- `dbinfo()` dbspace output enhancement

SQL features - audit to syslog()

- This feature enables `onlog` output to be written to the `syslog()` function for consolidation and integration with the rest of the enterprise auditing information
  - A feature primarily driven by customers in the financial industries

- This feature is only available for *nix platforms, not Windows

- Using this feature will require coordination with, and work done by, the O/S administrators to configure the `syslog` daemon and other interfaces requiring `root`-level permissions
  - Be aware that different O/Ss have different `syslog` names (e.g. `syslog` or `syslogd`) and functionality
    - You have to tailor the configuration to each O/S

- With this functionality, the `ADTMODE $ONCONFIG` parameter has been deprecated

- Full disclosure — I have limited to no experience with auditing

SQL features - audit to syslog()

- To support this functionality, `onaudit` has new or modified flags
  - The new flags use a capitalized flag and require a parameter value
    - The heritage `-l` (lowercase “el”) flag (and values) has been deprecated and replaced with `-L`, `-S` and `-A` options
  - `h` - Print help message and exit, new flag!
  - `q` - Suppress banner line, new flag!
  - `d` - sets the output to single rather than double-spaced by omitting the blank line between each result line, new flag!
  - `A` - Automatically audit DBSA actions
    - Either `on` or `off`
  - `E` - Enable or disable audit to `syslog()`
    - `0` (zero) disable, default
    - `1` (one) enable

SQL features - audit to syslog()

- To support this functionality, `onaudit` has new or modified flags
  - The new flags use a capitalized flag and require a parameter value
    - The heritage `-l` (lowercase “el”) flag (and values) has been deprecated and replaced with `-L`, `-S` and `-A` options
  - `F` - Facility - a `openlog()` term that precedes the `syslog()` call for message grouping and routing based on priority
    - Recommend using `LOG_USER`, `LOG_LOCAL0 .. LOG_LOCAL7` and a limited number of other identifiers
  - `I` - Identifier for messages - the instance name (default) or other explicitly set identifier
    - Is written into the audit log so the origin of the log record is known
  - `L` - Enable or disable classic auditing
  - `O` - Options for `openlog()` - typically is not set or used
  - `P` - Priority - used along with `Facility` to group and send messages to the correct repository based on importance
    - There are a number of priority values but `LOG_INFO` should be considered as the standard
  - `S` - Automatically audit DBSSO actions
    - Either `on` or `off`

SQL features - audit to syslog()

- The `onshowaudit` utility can now display audit information written to `syslog()` repositories
  - Use the `-j` *`pathed_location_to_file`* option
  - Be aware though - the `syslog()` function adds information to the start of an audit line that the classic Informix audit functionality does not
    - *Typically*, this data isn't pipe deleted so it *shouldn't* interfere with `onshowaudit` output processing
      - Your mileage may vary of course! 😁
  - In addition, the `syslog()` functionality may use a different timezone than the Informix instance making it difficult to find the relevant records

SQL features - temp space usage

SQL features - temp space usage

- The ability to look at temp space utilization has been available for a while
  - It involved using a rather convoluted SQL statement
  - That statement has been converted to a `sysmaster` view - `syssessiontempspaceusage`
    - For example:

```
select * from syssessiontempspaceusage
```

```
sid              46
flags            0x00000922
partition        0x00800002
table            my_stores:informix:my_temp
allocated_pages  8
```

  - Obviously with the session ID in the output, you can filter by it
    - If you can guess at the temp table name you can use that to filter with as well

- This statement returns both explicit and implicit temp space usage

SQL features - UDR last accessed

SQL features - last accessed

- Customers have wondered whether UDRs created in instances are being used
  - Until now, one way to find out was to drop one and see what breaks
    - *Not* the best option 😱

- The sysmaster `sysprc` and `sysdsc` tables contain information about the all UDRs (system and end-user created) in the instance
  - These tables now contain a last accessed timestamp for the UDR

`sysprc`

prc_hashno	0
prc_chainno	0
prc_id	632
prc_refcnt	0
prc_delete	0
prc_hits	1
prc_last_access	2021-04-16 09:49:47
prc_heapptr	1214793784
prc_heapsz	1
prc_servername	inst_1
prc_dbname	sysadmin
prc_ownername	
prc_name	check_backup

`sysdsc`

dis_hashno	25
dis_chainno	1
dis_id	0
dis_refcnt	0
dis_delete	0
dis_hits	1
dis_last_access	2021-04-16 09:49:50
dis_heapptr	1221747768
dis_heapsz	1
dis_servername	
dis_dbname	tpch
dis_ownername	informix
dis_name	sysdistrib.colno

SQL features - `dbinfo()` partnum enhancement

SQL features - dbinfo() partnum

- Customers have been using the `dbinfo()` function to interrogate the instance and using dbspace `partnum` as a parameter
  - Up to a certain point, the scripts have worked successfully
  - In other cases, the function “failed” with -727 errors
    - In truth the function didn’t fail, the `partnum` was invalid and the function reported as such

- Development wanted to create a way to gracefully return an error message on an invalid `partnum` but at the same time not change how `dbinfo()` functioned

- The result is a new environment parameter - `DBINFO_DBSPACE_RETURN_NULL_FOR_INVALID_PARTNUM`
  - Yes, it’s a long one! 🤣☺️
    - But you only have to set it once . . . . . if you set it at all

SQL features - dbinfo() partnum

- The variable can be set
  - At the session level
  - At the client level for all sessions
  - At the instance level (before starting the instance) for all connected sessions

- The variable values are
  - `0` (zero) - off, default `dbinfo()` behavior
  - `1` (one) - return null for invalid `partnum`, no -727 error
    - If a NULL `partnum` is passed to the function, a -727 error is returned however
    - In truth, almost any positive integer value can be used to set this variable

Questions

IBM INFORMIX V.14.10.XC6 - ER PERFORMANCE UPDATES

v.1b

Agenda

- What's new in ER with xC6
- Migrating to xC6

ER performance updates

- A significant amount of work went into improving the performance of ER
  - It touches all aspects - capture, queuing, transport, receive and apply phases of ER
  - Most of it is behind the scenes and not readily seen by you
    - In other words, there aren't any new, cool configuration parameters to set
    - There are a few things to change which is covered later
- Key thing to note — these improvements are only available between xC6 nodes
  - Nodes on earlier versions will not see these improvements

ER performance updates

- So how good is it??
  - In a series of benchmarks utilizing 80 million rows:
    - <xC6 - 2 hours to load
    - xC6 - 8 minutes
      - About a 15x improvement but should expect about 5 - 10x since so many other variables may affect throughput
  - Previously, ER could sustain about 2,000 transactions/sec throughput rate, in xC6 it's about 10,000 trans/sec
  - `cdr sync` performance is at least 4x faster
    - Unfortunately `cdr check` is still about the same
- As always, your mileage may vary! 😁😱😁

ER performance updates

- As mentioned there are no new magic bullets but there are new tuning guidance and recommendations

- First, the transport layer has changed within xC6
  - Instead of the *`nif`* mechanism, the SMX protocol is being used
  - This enables many improvements including *parallelism* throughout the transport layer
    - The `cdr_nifsend` thread could be, and was blocked based on various network conditions, SMX isn't
  - There is nothing you need to do to enable this, it is the default protocol between xC6 and later instances

  - NOTE: If any target is <xC6, the legacy *`nif`* communication mechanisms are used with that target

ER performance updates

- There is specific guidance on how to tune the SMX configuration parameters for ER
  - `smx_numpipes` - 2 or more per target node
    - You may need more depending on the transaction volume to all the targets
      - Testing indicates 7 or more *may* be appropriate based on transaction volume, transaction size and distance between the nodes
      - Monitor with the standard SMX `onstat` commands and tune based on queue length
  - `smx_ping_interval` - 30 seconds
    - Unlike the H/A cluster which needs a tighter tolerance for failure detection (10 seconds default), ER can use more relaxed standard
      - This value reduces the overhead in the communication component
  - `smx_ping_retry` - 6
    - This is the default value, leave it alone
  - For compression on the wire, use `smx_compress` instead of `cdr_nifcompress`
  - For encryption, TLS (`onsocssl`) is best but `encrypt_smx` can be used
    - If you don’t want to manage keys

ER performance updates

- Let's look at the data queuing and apply components
  - `cdr_queuemem` - configurable between 128 MB and 4GB, recommended 256 MB to 1 GB
    - Used to determine how much memory is used for the send and receive queues
      - Performance is about the same across this range
        - Use what you need, don't just set it high and waste resources
    - Tuning note for xC5 and earlier — performance degradation starts to occur when this value is greater than 128 MB
      - There is significant negative impact when it exceeds 256 MB
  - `cdr_evalthreads` - 0,7
    - Used to enable parallelism in the grouping threads, the old tuning guidance was CPUVPs +2
      - Testing showed tuning above 20 had a negative impact on performance
      - 7 seems to be the optimal number but should be monitored and modified carefully

ER performance updates

- A significant change occurred in the send queue mechanism

- Previously,
  - Transaction header information was stored on the `cdr_qhdr_dbspace`
    - This was deprecated in Informix v.12.10
    - Now the information is stored in the `cdr_dbspace` dbspace along with the `syscdr` database
  - Transaction data is still stored in SLOBspaces identified with `cdr_qdata_sbspace`
  - When transactions were flushed, I/O occurred to both spaces

- In xC6, the transaction header structure is expanded to include data as well !!
  - The data must be <26 KB to be stored with the header
  - The net impact, for small transactions, is 1/2 the I/O overhead for flushing small transactions to targets
    - This also reduces the workload through the SLOBspace APIs

ER performance updates

- A significant change occurred on the apply side as well — it's now parallelized for changes within the same table!!!
  - Including out-of-sequence writes

- Multiple transactions will be applied on the same target table provided there are no collisions within the key values
  - In other words, the changes are occurring to different rows

- But what about transaction acknowledgements? Wont things get messed up?
  - There is a new "`parallel_apply`" table that tracks what is written where
  - When all the updates for a transaction are applied
    - It sends a message to the acknowledgment mechanism which is still single threaded and replies in LSN order
      - It then replies to the sender that the transaction is completely applied
  - This mechanism supports replication and instance failure conditions
    - If, following a restart, transaction information is resent because it hasn't been ack'ed, this table controls what still needs to be applied on the target to complete the transaction

ER performance updates

- A word before continuing — parallel apply ** may ** affect some target side processing particularly if there are apply triggers on the table
  - There is some “voodoo” that can be applied to stop this
  - At this time, contact technical support for further guidance

ER check utility

- A new consistency check utility has been added with xC6
  - It compares metadata about servers, repls and replsets between a “reference” or master node and one or more target nodes
    - To ensure objects are identical between the nodes

```
cdr check catalog [connect options] [--master | -m]=ref_svr
[targ_svr | [—all |-a]] [--verbose | -v]
```

    - What’s the difference between verbose and non-verbose?
      - Just more output details
    - The utility will either return a `0` (zero) result code if all is well or one of the following `cdr` utility numeric error codes: `1`, `5`, `21`, `37`, `48`, `53`, `61`, `62`, `99`, `121`, `193`, `194`, `195`, `205`

ER check utility

- For example, checking the `inst_5` node in non-verbose mode

```
Inst_4: cdr check catalog -m g_inst4 g_inst5

Verifying server definitions...

Server definitions...OK


Verifying replicate definitions...

Replicate definitions...OK


Verifying replicate participant definitions...

Replicate participant definitions...OK


Verifying replicate participants...

Replicate participants...OK


Verifying replicate set definitions...

Replicate set definitions...OK


Verifying replicate set participants...

Replicate set participants...OK

Inst_4:
```

ER check utility

- For example, checking the `inst_5` node in verbose mode

```
Inst_4: cdr check catalog -m g_inst4 g_inst5 -v

Verifying server definitions...
Server definition matched for g_inst4 on g_inst4 and g_inst5
Server definition matched for g_inst5 on g_inst4 and g_inst5
Server definition matched for g_inst6 on g_inst4 and g_inst5
Server definitions...OK

Verifying replicate definitions...
Replicate definition matched for upd_any_templ_g_inst4_1_1_customer between g_inst4 and g_inst5
Replicate definition matched for upd_any_templ_g_inst4_1_2_orders between g_inst4 and g_inst5
Replicate definition matched for upd_any_templ_g_inst4_1_3_manufact between g_inst4 and g_inst5
Replicate definition matched for price_book between g_inst4 and g_inst5
Replicate definition matched for region_move between g_inst4 and g_inst5
Replicate definitions...OK

Verifying replicate participant definitions...
Participant 104 definition matched for upd_any_templ_g_inst4_1_1_customer between g_inst4 and g_inst5
Participant 105 definition matched for upd_any_templ_g_inst4_1_1_customer between g_inst4 and g_inst5
Participant 106 definition matched for upd_any_templ_g_inst4_1_1_customer between g_inst4 and g_inst5
Participant 104 definition matched for upd_any_templ_g_inst4_1_2_orders between g_inst4 and g_inst5
Participant 105 definition matched for upd_any_templ_g_inst4_1_2_orders between g_inst4 and g_inst5
Participant 106 definition matched for upd_any_templ_g_inst4_1_2_orders between g_inst4 and g_inst5
Participant 104 definition matched for upd_any_templ_g_inst4_1_3_manufact between g_inst4 and g_inst5
Participant 105 definition matched for upd_any_templ_g_inst4_1_3_manufact between g_inst4 and g_inst5
Participant 106 definition matched for upd_any_templ_g_inst4_1_3_manufact between g_inst4 and g_inst5
Participant 104 definition matched for price_book between g_inst4 and g_inst5
Participant 105 definition matched for price_book between g_inst4 and g_inst5
Participant 106 definition matched for price_book between g_inst4 and g_inst5
Participant 104 definition matched for region_move between g_inst4 and g_inst5
Participant 105 definition matched for region_move between g_inst4 and g_inst5
Participant 106 definition matched for region_move between g_inst4 and g_inst5
Replicate participant definitions...OK

Verifying replicate participants...
Replicate participants...OK

Verifying replicate set definitions...
Replicate set definition matched for upd_any_templ between g_inst4 and g_inst5
Replicate set definitions...OK

Verifying replicate set participants...
Replicate set participants...OK
Inst_4:
```

ER check utility

- Some things the utility doesn't capture
  - A repl is disabled

```
Inst_4: cdr check catalog -m g_inst4 g_inst5

Verifying server definitions...

Server definitions...OK


Verifying replicate definitions...

Replicate definitions...OK


Verifying replicate participant definitions...

Replicate participant definitions...OK


Verifying replicate participants...

Replicate participants...OK


Verifying replicate set definitions...

Replicate set definitions...OK


Verifying replicate set participants...

Replicate set participants...OK
```

```
informix@ifmx-svr:~/la
File  Edit  View  Search  Terminal  Help
STATE:              Active ON:g_inst5
CONFLICT:           Always Apply
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores:informix.my_prices
OPTIONS:            transaction,fullrow
REPLID:             6815754 / 0x68000a
REPLMODE:           READ-ONLY ON:g_inst5
APPLY-AS:           INFORMIX ON:g_inst5
REPLTYPE:           Master

REPLICATE:          region_move
STATE:              Inactive ON:g_inst5
CONFLICT:           Always Apply
FREQUENCY:          immediate
QUEUE SIZE:         0
PARTICIPANT:        stores:informix.local_stuff
OPTIONS:            transaction,fullrow
REPLID:             6815756 / 0x68000c
REPLMODE:           SEND-ONLY ON:g_inst5
APPLY-AS:           INFORMIX ON:g_inst5
REPLTYPE:           Master

Inst_5:
```

ER check utility

- Some things the utility doesn't capture
  - A node is disabled

```
Inst_5: cdr disable server g_inst5
Executing the cdr disable server command for server g_inst5 at server g_inst5 ... PASSED
Executing the cdr disable server command for server g_inst5 at server g_inst6 ... PASSED
Inst_5:
```

```
Inst_4: cdr check catalog -m g_inst4 g_inst5

Verifying server definitions...

Server definitions...OK


Verifying replicate definitions...

Replicate definitions...OK


Verifying replicate participant definitions...

Replicate participant definitions...OK


eplicate participants...

articipants...OK


Verifying replicate set definitions...

Replicate set definitions...OK


Verifying replicate set participants...

Replicate set participants...OK

Inst_4:
```

ER check utility

- Some things the utility doesn't capture
  - A template that is defined across the cluster but not realized

```
Inst_4: cdr check catalog -m g_inst4 g_inst5

Verifying server definitions...

Server definitions...OK


Verifying replicate definitions...

Replicate definitions...OK


Verifying replicate participant definitions...

Replicate participant definitions...OK


Verifying replicate participants...

Replicate participants...OK


Verifying replicate set definitions...

Replicate set definitions...OK


Verifying replicate set participants...

Replicate set participants...OK

Inst_4:
```

```
PARTICIPANT:      :.
OPTIONS:          transaction,ris,ats,fullrow,PendingSync
REPLID:           6815758 / 0x68000e
REPLMODE:         OTHER () ON:g_inst5
APPLY-AS:         OWNER ON:g_inst5
REPLTYPE:         Master

REPLICATE:        upd_any_templ_g_inst4_21_2_orders
STATE:            Inactive ON:g_inst5
CONFLICT:         Timestamp
FREQUENCY:        immediate
QUEUE SIZE:       0
PARTICIPANT:      :.
OPTIONS:          transaction,ris,ats,fullrow,PendingSync
REPLID:           6815759 / 0x68000f
REPLMODE:         OTHER () ON:g_inst5
APPLY-AS:         OWNER ON:g_inst5
REPLTYPE:         Master

REPLICATE:        upd_any_templ_g_inst4_21_3_manufact
STATE:            Inactive ON:g_inst5
CONFLICT:         Timestamp
FREQUENCY:        immediate
QUEUE SIZE:       0
PARTICIPANT:      :.
OPTIONS:          transaction,ris,ats,fullrow,PendingSync
REPLID:           6815760 / 0x680010
REPLMODE:         OTHER () ON:g_inst5
APPLY-AS:         OWNER ON:g_inst5
REPLTYPE:         Master

Inst_5: cdr list templ

TEMPLATE            DATABASE            TABLES
============================================
upd_any_templ       stores              informix.customer
                    stores              informix.manufact
                    stores              informix.orders
Inst_5:
```

Migrating ER to xC6

ER performance updates

- In order to take advantage of these new features, specific steps must followed when migrating to xC6
  - It does require an outage on each node

- Before upgrading
  - Make sure all the ER queues are empty and the log replay position is at current
    - Use the `onstat -g rqm` command to check the queues and the `onstat -g ddr` command to check the replay position
  - Stop ER on the node with a `cdr stop` command

- Upgrade the binary to xC6 and restart the instance

- Use the `cdr cleanstart` command to restart ER on the node
  - This forces several tables to be dropped and recreated in the new schema
  - These contain transaction information thus it's important they are clean and clear before continuing

Questions

Part 2 of the series, July 13

Grazie — ITALIAN
धन्यवाद — HINDI
Merci — FRENCH
ありがとうございました — JAPANESE
Obrigado — BRAZILIAN PORTUGUESE
多谢 — SIMPLIFIED CHINESE

Thank You

多謝 — TRADITIONAL CHINESE
Gracias — SPANISH
Спасибо — RUSSIAN
நன்றி — TAMIL
ขอบคุณ — THAI
Danke — GERMAN
شكراً — ARABIC