



A brief overview of what's new in IBM Informix v.14.10

Carlton Doe
Informix Technical Team
cdoe@us.ibm.com

Scott Pickett
Informix Technical Team
spickett@us.ibm.com



A brief overview of

- What's new in replication within IBM Informix v.14.10
- FC8 enhancements

Carlton Doe
Informix Technical Team
cdoe@us.ibm.com

Agenda

- H/A and ER improvements in v.14.10
 - H/A cluster performance enhancements
 - Asynchronous post commit triggers
 - Offline conversation for H/A secondaries
 - `cdr migrate server` enhancements
 - Flow control statistics
 - ER performance
 - ER check utility
 - Required FC6 upgrade process
 - Backup from RS secondaries
- Overview of features in v.14.10.FC8
 - Critical migration / upgrade procedures
 - H/A cluster
 - IHQ
 - Internal Java updates
 - Additional Read Ahead enhancements
 - IHQ 2.1
 - Announcements

A quick bit of news

- Getting access to CSDK and JDBC packages has historically been a problem
- All of the available CSDK and JDBC packages have been migrated to the Informix Trials and Download site
 - Will reside along with the Developer Edition and time-limited engine packages

https://www.ibm.com/resources/mrs/assets?source=ifxids&lang=en_US

- About 255 packages from 4.10 and 4.50 were moved over
- The International Language Support module was uploaded as well (v.3.50.MC7)

H/A and ER enhancements in v.14.10

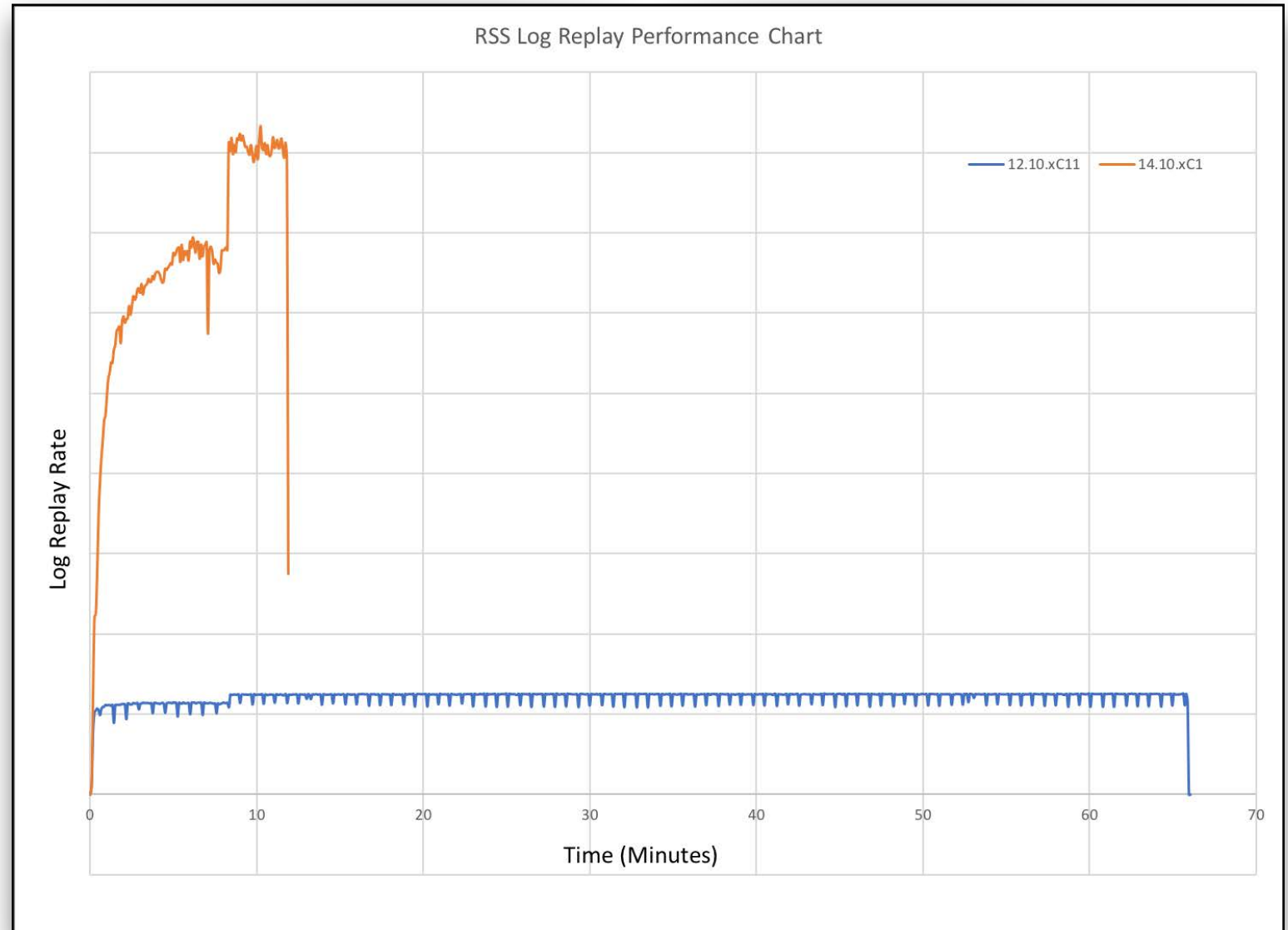
Performance enhancements to the H/A cluster

Performance and monitoring, H/A cluster

- With IBM Informix v.14.10, the RS node apply process was completely re-written and optimized for *much, much* higher performance and throughput
 - We're now very close to being limited by the hardware and its processing bandwidth and throughput
 - How fast are we?
 - See next slide
- Similar work was performed on the HDR apply process earlier in IBM Informix v.12.10
- This enhancement applies to all log replay and log apply operations, not just RS

Performance and monitoring, H/A cluster

- A comparison of RSS throughput between IBM Informix v.12.10 and IBM Informix v.14.10
 - 8 minute high intensity workload on the primary



Performance and monitoring, H/A cluster

- There are numerous benefits to customers with these improvements
 - Near zero replication latency between H/A cluster nodes
 - Enables customers to load balance applications across the cluster rather than concentrating high DML workloads on the primary
 - Customers can feel even more confident in the ability of the H/A cluster technology to meet their data protection requirements and design
 - With a 5x improvement in crash recovery, helps the cluster meet the recovery time objectives in the event of a cluster / server crash
 - With a 5x improvement in logical log restore performance, significantly reduces the time required for the logical restore portion of an instance / dbspace restore

Performance and monitoring, H/A cluster

- With these enhancements, there are `onstat` and `$ONCONFIG` enhancements and changes
 - It's not a new release of IBM Informix without something being added to `onstat`!!
- When executing the `onstat -g laq` command (print secondary queues) in “repeat mode” (`-r [timevalue]`), the log replay and latency rate is now available
 - The latency value is calculated by comparing the commit timestamp from the primary and the apply timestamp on the secondary
 - For this value to be accurate, the system clocks must be synced

```
Secondary Apply Queue:      Total Buffers:12 Size:2048K Free Buffers:0
Log Recovery Queue:        Total Buffers:12 Size:20480K Free Buffers:0
Log Page Queue:           Total Buffers:512 Size:4K Free Buffers:1
Log Record Queue:         Total Buffers:1000 Size:16K Free Buffers:1
```

```
Transaction Latency: 1 seconds
Apply rate: 347887.41 recs/sec
```

Performance and monitoring, H/A cluster

- New `$ONCONFIG` parameters, most of which are only of value to the most extreme workloads
 - `SEC_APPLY_POOLTIME`
 - Length of time in microseconds a log replay thread polls for work before yielding
 - `SEC_LOGREC_MAXBUFS`
 - The number of log buffers the secondary uses to replay logical log records
 - Each buffer is 16 KB
 - `RSS_NONBLOCKING_CKPT`
 - Enable non-blocking checkpoints on the secondary
 - Typically not needed
 - `SEC_DR_BUFS`
 - The number of log replication buffers for H/A cluster use
 - These buffers are the same size as `LOGBUFF`
 - `OFF_RECOVERY_THREADS` recommendations
 - 5 is still the recommended number but can be configured to 5, 7, 11 or 23
 - `DELAY_APPLY`
 - New parameter value: -1
 - Ignore the configured time value, stage the logical log and apply it immediately

Asynchronous post-commit trigger

ER enhancements - replication to SProc

- The ability to replicate to a *stored procedure*, not a table!!
 - Otherwise referred to as a asynchronous post-commit trigger
- There are numerous use cases for this including
 - Realtime analysis of data as it's being loaded
 - Can update consumers such as sales reports or leaderboards through asynchronous triggers
 - Replicate data to a table that doesn't have a primary key
 - The SProc can insert / update / delete data in any table
 - Transform data before loading into a table
 - The SProc can add / delete / modify columns or their values before sending it to the target
 - Create a CDC type of insert mechanism into external data sources
 - The SProc can output data to a staging table or file to be picked up by another process

ER enhancements - replication to SProc

- Similar to “regular” loopback replication
 - A target table must exist
 - This table can be in the same instance / database or somewhere else
 - *Difference* — the target table can be the source table!!
 - The normal ER transaction processing occurs
 - Data is manipulated
 - The “snoopy” picks it up based on rule evaluation
 - Means the `where` clause of the repl definition determines when replication to the SProc occurs
 - Data is staged in the replication queues
- However, instead of the replicated data being applied to the target table, it is sent to the SProc for it to operate on
 - There are specific SProc signature requirements for the SProc to function properly

ER enhancements - replication to SProc

- There are three new options to the `cdr define replicate` command

`--splname=name` the name of the stored procedure

- The SProc must exist on all ER participants
- There is a specific signature for this SProc

`--jsonsplname=name` the name of the stored procedure

- The SProc must exist on all ER participants
- In this case, the input argument is a JSON document with a specific format

- **NOTE:** `splname` and `jsonsplname` are mutually exclusive

`-cascaderepl=[y | n]`

- Only for customers with existing ER repls using post commit triggers
- Should the SProc data operations be replicated back to the source as well?
 - Could end up in an infinite loop condition if set incorrectly

ER enhancements - replication to SProc

- In addition to the new keywords, there is another critical flag to use when defining the repl:
`--serial`
 - Normally ER will attempt to group and parallelize operations to tables
 - It will look at each operation and see if several operations can be concatenated together to reduce data in the queues and reduce network traffic
- When replicating to a SProc, you want the procedure to execute on each row in order
 - This is particularly important if the SProc is executing streaming aggregations
 - Without this option, there is a risk of a race condition if multiple instances of the SProc are fired by different transactions
 - Incorrect results could be returned because values from peer transactions are ignored
- For these reasons, turn off parallelism for any SProc repls

Off-line conversion support for HDR and RS secondaries

Off-line conversion support

- Informix v.14.10.xC4 introduced off-line secondary conversion for HDR and RS secondary instances for major upgrades
 - You'll still have to take the instances off-line to convert but you won't have to rebuild the secondary instances
- This functionality supports in-place, offline upgrades from 11.70.xC1 to 14.10.xC4 or later
- Please note — this is a one way operation, reversion to an earlier Informix version (even the original secondary version) is NOT supported
- Enabling `CONVERSION_GUARD` for all instances is ***strongly*** recommended

Off-line conversion support

- What is the workflow for this process?
 - Disable FOC so a secondary instance is NOT promoted to primary
 - After ensuring `CONVERSION_GUARD` is enabled for all instances, shut down the primary instance
 - Updatable secondary instances will be blocked from attempting any DML operations
 - All secondary read operations will continue without being affected
 - Install the new Informix binary and make any `$ONCONFIG` changes needed for new functionality
 - Restart the primary
 - This instance will go through the upgrade process
 - All behind the scenes instance changes are logged in the logical logs!!!
 - If the upgrade fails, use `onrestorererept` to roll back the conversion changes, fix the issues then try again
 - So you know, during the conversion / upgrade process for the primary or secondary nodes, no end-user connections are allowed until the instance is fully converted

Off-line conversion support

- What is the workflow for this process?
 - With the primary successfully upgraded, the secondary instances will NOT re-connect because of the version mis-match checking
 - The secondary instances will still think the primary is off-line
 - Turn the first secondary off-line
 - Install the new Informix binary on the secondary, make any `$ONCONFIG` changes, and restart the instance
 - This instance will go through the upgrade process
 - Upgrade changes from the primary instance logical log files are sent to the secondary so they are applied
 - If the upgrade fails, use `onrestorer rept` to roll back the conversion changes, fix the issues then try again
 - Once the upgrade process has completed, execute an `onmode -c` command on the primary to force a cluster checkpoint and commit the changes on the secondary
 - The upgraded secondary will re-connect to the cluster primary and be fully functional

Off-line conversion support

- What is the workflow for this process?
 - Continue this process for each HDR and RS secondary in the cluster
 - Turn FOC back on
 - Congratulations, you've upgraded the cluster without having to rebuild the secondary instances!
- But what if I have a SD secondary instance?
 - Turn it off before making any changes to the cluster or other secondary instances
 - Once the other instances are upgraded and online
 - Install the new binary on the SD secondary node(s), make any `$ONCONFIG` changes, and turn them on
 - Since SD secondary instances use the same disks as the primary, no conversion is required

cdr migrate server enhancements

`cdr migrate server` enhancements

- You can execute `cdr migrate server` operations in two modes — *static* or *dynamic*
 - `static` mode, or `offline` means the source instance is blocked and the data is migrated as though it was a backup and recovery operation
 - There is no need to worry about referential constraint violations
 - `dynamic` mode or `online` migration means the source instance continues to process transactions during the migration process
 - There is a potential for RI violations to occur during the data migration phase

cdr migrate server enhancements

- The actual data loading process *for v.14.10.FC3 and earlier* has several steps to it
 - Step 1:
 - Migrate and create the database schema on the target
 - Indexes, PKs and constraints (unique or referential [FK]) are NOT created
 - The tables are created in `raw` mode
 - Step 2:
 - Start multiple, parallel jobs for data loading and index creation
 - Uses the Informix v.14.10 `insert into .. select * from ..` distributed SQL functionality
 - When the table is loaded, indexes, PKs and unique constraints are built
 - Step 3:
 - Referential constraints (FKs) are created and verified
 - Once the data is loaded, a data sync phase is executed to verify the data sets
- Because of the parallel load process, it's possible for RI constraints to fail causing the migration process to fail

`cdr migrate server` enhancements

- With v.14.10.FC4, the `create_schema_loaddata_nori` clause was added to stop RI building in `dynamic` migrations
- The `add_ri` phase was added to rebuild RI after the `data_sync` phase of `dynamic` migrations
- Together, these clauses help resolve data sync and RI problems that arise from multiple parallel loads occurring from an active instance

`cdr migrate server` enhancements

- With v.14.10.FC4, a new phase was added to `cdr migrate server` — `add_replcheck`
- An optional phase, it is not automatically executed in a migration
 - If it is executed manually, it should be executed first on the source instance before any other phases
 - By the way — if the target is empty / new, there's no point to executing this since all data must be copied to the target
 - That said, the `replcheck` column can be useful to speed up the data verification task if you decide to keep ER active after the data migration process
- This phase creates a shadow `ifx_replcheck` column (`bigint`) and a composite unique index on all source, and, by extension, target tables
 - WARNING!!! Adding the column is a *slow alter* requiring an outage on the table

`cdr migrate server` enhancements

- The value in the `ifx_replcheck` column varies depending on whether the row is a new insert / pre-existing at command execution or modified after command execution
 - New inserts or pre-existing — the column contains a checksum of the row's values
 - Modified rows — the column contains a version related value
 - Both values incorporate the value for the `cdr group ID`
 - As defined in `SQLHOSTS`
- The `add_replcheck` index incorporates the `ifx_replcheck` shadow column and the `PK / UI / ERKey` columns
 - There are no options to control how this index is built (dbspaces, fragmentation, etc.)
 - However, with the “no execute” and print options, you can get the `add_replchck` phase commands to be executed, modify the index creation operations and execute the script manually
- Why do this? It dramatically increases the performance of the `sync_data` phase

Flow control statistics for SD and RS secondaries

Flow control statistics for SD and RS secondary instances

- In Informix v.11.50.xC6 (for `RSS_FLOW_CONTROL`) and Informix v.11.70 (for `SDS_FLOW_CONTROL`), flow control was introduced to help these instances stay current with the cluster primary
- It was possible for the RS / SD secondaries to fall behind in applying / recognizing committed transactions
 - Either because of slow network transmission, insufficient power on the secondary to handle the workload, or the transaction volume
 - The further behind these nodes became, the less useful they were for supporting connected sessions or being a potential failover target
 - If the secondaries fell too far behind, it could cause a logical log rollover condition on the primary resulting in suspension of instance activities until the secondary instance(s) caught back up
- The flow control parameter provided a way to throttle primary processing to help the secondary nodes to keep up

Flow control statistics for SD and RS secondary instances

- With the improved roll forward technology introduced at the end of Informix v.12.10 (for HDR) and Informix v.14.10.FC1 (for RS secondaries), you probably won't need flow control any more
 - The apply rate is insanely quick!
- In Informix 14.10.FC4 more flow control statistical information is available in the `onstat -g [rss | sds] verbose` output
 - If applications are experiencing processing delays, this information can help diagnose whether or not flow control is the issue

Flow control statistics for SD and RS secondary


- For example, in a quiet test environment with one SD and RS secondary instance
 - From the primary looking at the RS secondary

```
Local server type: Primary
Index page logging status: Enabled
Index page logging was enabled at: 2020/06/24 10:28:08
Number of RSS servers: 2

RSS Server information:

RSS Server control block: (nil)
RSS server name: inst_5
RSS server status: Defined
RSS connection status: Disconnected

RSS Server control block: 0x46f78d20
RSS server name: inst_4
RSS server status: Active
RSS connection status: Connected
RSS flow control:384/352
Log transmission status: Active
Next log page to send(log id,page): 41,126
Last log page acked(log id,page): 41,125
Last log page applied(log id,page): 41,125
Time of Last Acknowledgement: 2020-06-26.14:33:01
Pending Log Pages to be ACKed: 0
Approximate Log Page Backlog:0
Sequence number of next buffer to send: 443
Sequence number of last buffer acked: 442
Supports Proxy Writes: Y
Total number of delay(s): 14
Time of last delay: 2020-06-26.13:42:12
```



ER Performance in FC6

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 and later 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 FC6
 - 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
 - `smx_ping_interval` - 30 seconds
 - `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
 - `cdr_queuemem` - configurable between 128 MB and 4GB, recommended 256 MB to 1 GB
 - `cdr_evalthreads` - 0,7

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 was 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 quick word of caution — 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 in FC6

ER check utility

- A new consistency check utility was 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

HA cluster / ER FC6 migration

Conversion requirement

- In Informix 12.10.xC5, rolling upgrades of H/A clusters was introduced
 - From one fix pack to the next
- We said 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 FC6!

Conversion requirement

- For FC6, in 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
 - 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

ER FC6 migration

- In order to take advantage of the new ER features, specific steps must followed when migrating to FC6
 - 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 FC6 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 upgrading

Backup from RS secondary

RS Backups

- The ability to backup an H/A cluster from a node other than the cluster primary has been a customer request dating back many years
- Starting with Informix v.14.10FC6 it is now possible — from an RS secondary
- There were significant changes to this functionality in FC8
- To enable RS backups, the `BAR_SEC_ALLOW_BACKUP $ONCONFIG` parameter must be set
 - It was not included in `$ONCONFIG` in FC6 but now is
 - It is NOT dynamically tunable, an instance restart required
 - Potential values
 - 0 — RS backups are disabled
 - 1 (one) — “local” RS backup mode

RS backups

- Originally, there were caveats or conditions for this functionality to work
 - No unlogged objects in the instance
 - Non-logged SLOBspace
 - Raw table in a logged database
 - Non-logged database in an instance
 - Non-logged SLOB in the database, even if stored in a logged SLOBspace
- Set the then new `TEMPTAB_NOLOG` parameter to 2
 - Support for automatic switching of logged temp table support when a node moves from secondary to primary status
 - Prior to this setting, if a secondary node became primary, the original `NOLOG` setting persisted and logged temp table support was not available
 - Now it will change as the node's role changes from secondary to primary

RS backups

- To enable RS backups with non-logged objects, set the `ARCHIVE_UNLOGGED_OBJECTS` parameter and restart the instance
 - FYI - it was originally improperly documented as (`ARCHIVE_NOLOG_OBJECTS`)
- Potential values
 - 0 (zero) — backups blocked if non-logged objects exist
 - 1 (one) — quiet mode, backups allowed though a general message that the backup contains non-logged objects and may not be fully consistent is written to the message log
 - 2 (two) — verbose mode, backups allowed and each non-logged object found is written to the instance log along with the general message that the backup contains non-logged objects and may not be fully consistent

RS backups

- The `ARCHIVE_UNLOGGED_OBJECTS` parameter applies to `ontape` and `ON-Bar` backup operations
- My test instance (`inst_4`) has non-logged SLOBspaces as well as a raw table

```
Inst_4_8: onstat -d
IBM Informix Dynamic Server Version 14.10.FC8WE -- Updatable (RSS) -- Up 00:01:04 -- 234176 Kbytes
2022-09-06 13:31:32

Dbspaces
address          number  flags      fchunk  nchunks  pgsize  flags  owner  name
4596b028         1      0x801      1       1        2048   NL BA  informix rootdbs
45aa3de8         2      0x801      2       1        2048   NL BA  informix data_space_1
4596bdc8         3      0x801      3       1        2048   NL BA  informix log_space
4596cbf0         4      0x2001     4       1        2048   N TBA  informix work_space
45a86050         5      0x8801     5       1        2048   NLSBA  informix slob_space
45a86290         6      0xa001     6       1        2048   N UBA  informix slob_temp
6 active, 2047 maximum
```

```
Chunks
address          chunk/dbs  offset  size  free  bpages  flags
4596b268         1 1 0 50000 36590 0
4611f028         2 2 0 102400 100046 0
46120028         3 3 0 102400 32347 0
46121028         4 4 0 102400 102347 0
46122028         5 5 0 25600 23802 23802
46123028         6 6 0 102400 -1 -1
6 active, 32766 maximum
```

```
inst_4_8: dbschema -d stores -t rawtab
DBSCHEMA Schema Utility      INFORMIX-SQL Version 14.10.FC8
{ TABLE "informix".rawtab row size = 2 number of columns = 1
create raw table "informix".rawtab
(
  coll smallint
);
revoke all on "informix".rawtab from "public" as "informix";
```

Changes to RSS backups

- With the `ARCHIVE_UNLOGGED_OBJECTS` parameter = 0
 - Backup is blocked

```
Inst_4_8: ontape -s -L 0
Archive failed - (-83380) An archive checkpoint could not be completed in the secondary server.

Program over.
Inst_4_8:
Inst_4_8:

16:22:48 The storage space, 'slob_space', is preventing the backup on the
          secondary server.
16:22:49 (-83380) An archive checkpoint could not be completed in the secondary server.
```

Changes to RSS backups

- With the `ARCHIVE_UNLOGGED_OBJECTS` parameter = 1
 - Backup is allowed with a general message in the instance log

```
Inst_4_8: ontape -s -L 0
100 percent done.
File created: /opt/IBM/informix/backup/inst4_L0

Please label this tape as number 1 in the arc tape sequence.
This tape contains the following logical logs:

18

Program over.
```

```
15:55:24 WARNING: This archive contains unlogged objects such as raw tables
or BLOBspace blobs and is therefore incomplete because these objects
are not fully replicated. If restored from this archive these objects
will be internally inconsistent and must be recreated before they
can be used.
15:55:25 Level 0 Archive started on rootdbs, data_space_1, log_space, slob_space
15:55:26 Archive on rootdbs, data_space_1, log_space, slob_space Completed.
15:56:17 Logical Log 18 Complete, timestamp: 0xb3f9c.
15:56:23 Checkpoint Completed: duration was 0 seconds.
```

Changes to RSS backups

- With the `ARCHIVE_UNLOGGED_OBJECTS` parameter = 2
 - Backup is allowed with more detailed messages in the instance log

```
15:57:35 Checkpoint Statistics - Avg. Txn Block Time 0.000, # Txns blocked 0, Plog u
d 0
15:57:35 The storage space, 'slob_space', which contains
           partially-replicated objects, would normally prevent a backup on the
           secondary server.
15:57:35 The raw table 'stores':'rawtab' would normally prevent
           a backup on the secondary server.
15:57:35 These restrictions have been overridden by the user.
15:57:35 WARNING: This archive contains unlogged objects such as raw tables
           or BLOBspace blobs and is therefore incomplete because these objects
           are not fully replicated. If restored from this archive these objects
           will be internally inconsistent and must be recreated before they
           can be used.
15:57:36 Level 0 Archive started on rootdbs, data_space_1, log_space, slob_space
15:57:37 Archive on rootdbs, data_space_1, log_space, slob_space Completed.
```

RS backups

- Informix is now enforcing that the `LTAPEDEV` parameter is the same throughout the cluster
 - In the past, the primary could have one value and secondary another value
 - It allowed the creation of “cluster” backups from the secondary
 - Primary was set to `/dev/null` while RS had “real” values
 - The primary didn’t try to trigger a logical log backup
 - As a result, you needed to review the tape parameters if the node was promoted to primary to ensure backups actually occurred
 - Now, if they are different, the secondary will not come online

RS backups

- Regardless of the `ARCHIVE_UNLOGGED_OBJECTS` parameter value, restoring the backup prints messages in the instance log that the backup is not complete

```
Inst_4_8: ontape -r
Restore will use level 0 archive file /opt/IBM/informix/backup/inst4_L0. Press Return to

Archive Tape Information

Tape type:      Archive Backup Tape

16:43:21
Backup is incomplete, it contain RAW objects backup, restore it at your own risk
16:43:28
Backup is incomplete, it contain RAW objects backup, restore it at your own risk
16:43:28 Parameters user configured value was adjusted. (MAX_PRIORITY)
```

Questions



Overview of feature and enhancements in v.14.10.FC8

Critical migration processes for H/A cluster and IHQ

H/A cluster

H/A clusters with the Connection Manager

- With Informix v.14.10.FC7, it became possible to make `sysmaster` database changes during a regular fix pack upgrade
 - Development intends on using this to make minor enhancements and changes as needed
- With Informix v.14.10.FC8 some systemic changes occurred requiring rebuilding of the system databases
- The process explained herein allows you to migrate an H/A cluster with Connection Manager running FC8 while maintaining most database services
 - At a specific point
 - DDL operations from a secondary are blocked but DML operations are allowed
 - The primary goes into single user mode blocking all access to make systemic changes
- Note — trying to test this with Developer Edition will fail
 - When the new primary is migrating to FC8 and rebuilding the system databases, there are too many connections to the instance
 - Developer Edition only allows 25 connections at a time

H/A clusters with the Connection Manager

- Process for a successful in-place H/A cluster migration to FC8
 - Must be using FC7W1
 - Should install FC8 software in a different / separate directory
 - This has additional benefits for the IHQ migration discussed next
 - The process involves migration of the Connection Manager agents to FC8 *first*
 - It has the logic required to handle the `sysmaster` rebuild that must occur
 - Turn off, migrate each secondary to FC8 then restart
 - During this time DDL operations from the secondary are blocked though DML operations are allowed
 - When time to migrate the primary, it will go into single-user mode to make the required `sysmaster` upgrades
 - These are logged and shipped to the secondaries so they get updated to FC8 schemas
 - When primary is finished the `sysmaster` migration, it comes back online and CM agents as well as secondaries reconnect
 - There is interesting functionality with respect to the primary and HDR secondary during the failover
 - It allows for easier integration of the “down” node back into the cluster

H/A clusters with the Connection Manager

- For a complete explanation of the process with a walk through, see the FC8 Deep dive webcast on the Informix Community library section (replay and presentation) or the IIUG TechTalk You Tube channel (replay only)



Critical migration processes for H/A cluster and IHQ

IHQ

Migration / upgrade processes - IHQ

- With Informix V.14.10.FC8, IHQ moves its internal database from v.1.6.3 (FC7) to v.2.0+
 - The feature list of what's in IHQ v.2.0+ is covered later
- In the FC8 release, IHQ v.2.0 is distributed
 - IHQ development strongly recommends downloading and using IHQ v.2.1 instead
 - It is available on FixCentral
- After downloading, copy them into `$INFORMIXDIR/hq` and replace the existing server and agent files
 - This assumes FC7 and FC8 are installed in different directories

```
Inst_1_8: cp informixhq-server-2.1.0.jar informixhq-server.jar
Inst_1_8:
Inst_1_8: cp informixhq-agent-2.1.0.jar informixhq-agent.jar
Inst_1_8:
```

Fix Central

Help

Search all IBM for Fix metadata

tips

Filter search by - Informix Server

Download files using HTTPS

Information Management, Informix Server (14.10.FC8, Linux 64-bit,x86_64)

[Subscribe to support notifications](#)

Download files using your web browser

Click the download link next to each file to download it.

Order number:	424131650
Total size:	29.29 MB

[Show normalized list](#) | [Hide normalized list](#)

interim fix: informixhq-agent-2.1.0

informixhq-agent-2.1.0

The following files implement this fix.

[↓ informixhq-agent-2.1.0.jar \(4.19 MB\)](#)

interim fix: informixhq-server-2.1.0

informixhq-server-2.1.0

The following files implement this fix.

[↓ informixhq-server-2.1.0.jar \(25.1 MB\)](#)

Migration / upgrade processes - IHQ

- In the move to v.2.0+, the underlying H2 database is upgraded
 - The database stores IHQ user and other IHQ configuration information
 - When IHQ is started for the first time, the database files are created in `$INFORMIXDIR/hq`

```
rw-rw-r--r-- 1 informix informix 5521 Nov 30 10:47 agent_properties.ins
-rw-rw-r--r-- 1 informix informix 573440 Dec  2 12:29 h2db.mv.db
-rw-rw-r--r-- 1 informix informix      0 Nov 30 15:21 h2db.trace.db
-rw-r--r--r-- 1 informix informix 5522 Oct 23  2021 informixhq-agent-exam
```

- This database does NOT support in-place migration, a manual process is required
- BEST PRACTICE — install FC8 in a different directory than FC7 (or earlier), perform the IHQ migration

Migration / upgrade processes - IHQ

- If an in-place IHQ upgrade is attempted, either by
 - Copying the older H2 database into the FC8 directory and turning on IHQ
 - Performing an in-place Informix binary install and starting with the new binary on the older H2 database

It will fail

- You may possibly compromise the existing data in the database requiring recreating all IHQ administrative and other objects!
- The best workflow is to
 - Install FC8 in a separate directory as already mentioned
 - Download and copy the IHQ 2.1 server and agent `.jar` files to the FC8 directory
 - Overwrite the existing files
 - Copy the H2 database file into the FC8 directory
 - Move the IHQ properties files over
 - Execute the H2 database migration command discussed next
 - Start IHQ with new binary and database

Migration / upgrade processes - IHQ

- For a complete explanation of the process with a walk through, see the FC8 Deep dive webcast on the Informix Community library section (replay and presentation) or the IIUG TechTalk You Tube channel (replay only)
- The presentation Appendix contains instructions migration instructions if you have to do a true in-place upgrade of IHQ (install FC8 on top of the earlier version)
 - The appendix is part of the presentation available on the Informix Community library section

Internal Java update

Internal Java update

- The Java environment currently bundled with Informix has been upgraded to 8.0.7
 - The exception is HP-UX which is using 8.0.6.30
 - For FC9 it's 8.0.7.11 for all platforms except
 - HP-UX - 8.0.7.0
 - Solaris (all) - 8.0.7.10
- See the *Announcements* section later for news about the bundled JRE



Additional Read Ahead Enhancements

Additional Read Ahead Enhancements

- A quick refresher — in FC7 the `onstat -g rah` output was enhanced to include new sections
 - General information on current conditions
 - Requests in the queue
 - What each of the daemons are working on
 - Read ahead while in system recovery
 - Read ahead by partition / thread was segmented into categories: data reads, index reads etc.
 - Columns to calculate effectiveness were added as well

Additional Read Ahead Enhancements

- New in FC8 are two additional, totals columns for thread and partition activity
 - These can be used to gauge R/A effectiveness across the object

Partn_Pages				Total RA Pages	
#reqs	pagecnt	nios	eff	read	used
0	0	0	0	23	20
0	0	0	0	29	8
0	0	0	0	54	39
0	0	0	0	3	0
0	0	0	0	2	0
0	0	0	0	34	2
0	0	0	0	1	1
0	0	0	0	131	32
0	0	0	0	8	4
0	0	0	0	0	0
0	0	0	0	7	5
0	0	0	0	6	1
0	0	0	0	32	24
0	0	0	0	1	0
0	0	0	0	1	0
0	0	0	0	1	1

Total Read is the number of pages put into the cache

RA Pages Used is the number of pages used from the cache

Ideally you want these two numbers to be as similar as possible. Can use this to fine tune R/A parameters.

Additional Read Ahead Enhancements

- New in FC8 is the re-introduction of the `R/A threshold` field in the `AUTO_READAHEAD` config parameter as well as session environment variable
 - If the value is `< 50`, the next batch is requested later in the processing sequence allowing more of the existing batch to be used
 - `> 50`, the batch is requested earlier

```
Inst_5_8: onstat -g cfg full auto_readahead
```

```
IBM Informix Dynamic Server Version 14.10.FC8WE -- On-Line -- Up 00:00:20 -- 234176 Kbytes
2022-09-08 14:27:46
```

```
Configuration Parameter Info
```

id	name	type	maxlen	units	rsvd	tunable
85	AUTO_READAHEAD	STRUCT	513		*	*

```
default : 1,128
onconfig: 1,128,25
current : 1,128,25
```

Description:

Use the AUTO_READAHEAD conf...
automatic read-ahead mode d

number of pages read ahead at a time. Range: 4-4096. Default: 128.

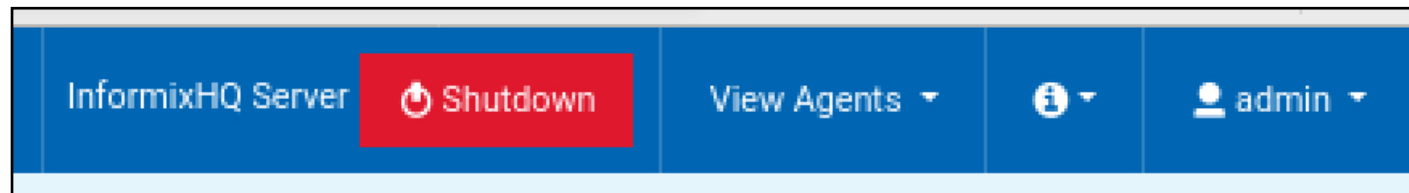
The third component, which is also optional, is a threshold--a whole number percentage (1-100). The next batch of pages will be requested once this percentage of the current batch remains unprocessed by the reader. Default: 50. `<50 == request next batch later.` `>50 == request next batch earlier.`



IHQ 2.1

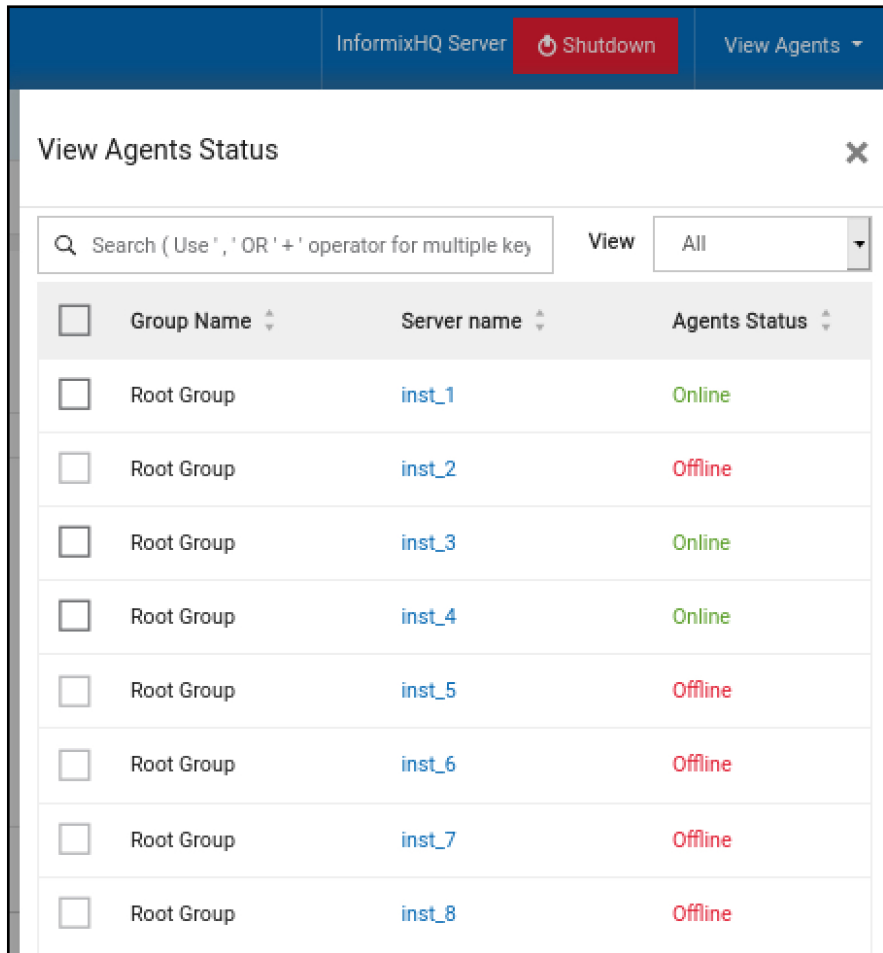
IHQ 2.1

- With the major systemic changes to the H2 database and the migration process discussed earlier, the IHQ tool has moved to v.2.0+ with a number of changes and bug fixes
 - As discussed in the IHQ migration section, FC8 distributes IHQ v.2.0 but development recommends upgrading to IHQ v.2.1
- To begin, the administration interface offers more options than previously available at the command line
- When logged in as an IHQ administrative user, two new options are available on the tool header:



IHQ 2.1

- You can filter which agents are displayed in the **View** dropdown

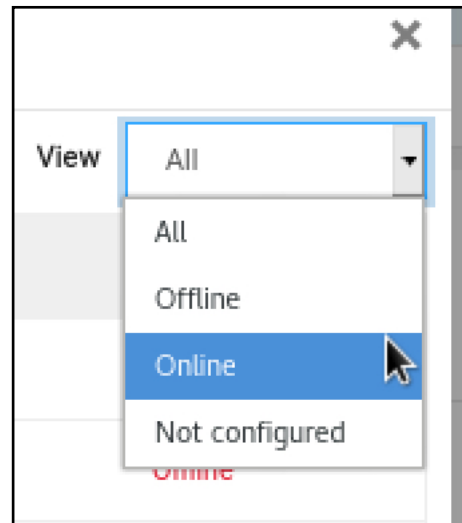


InformixHQ Server Shutdown View Agents ▾

View Agents Status ×

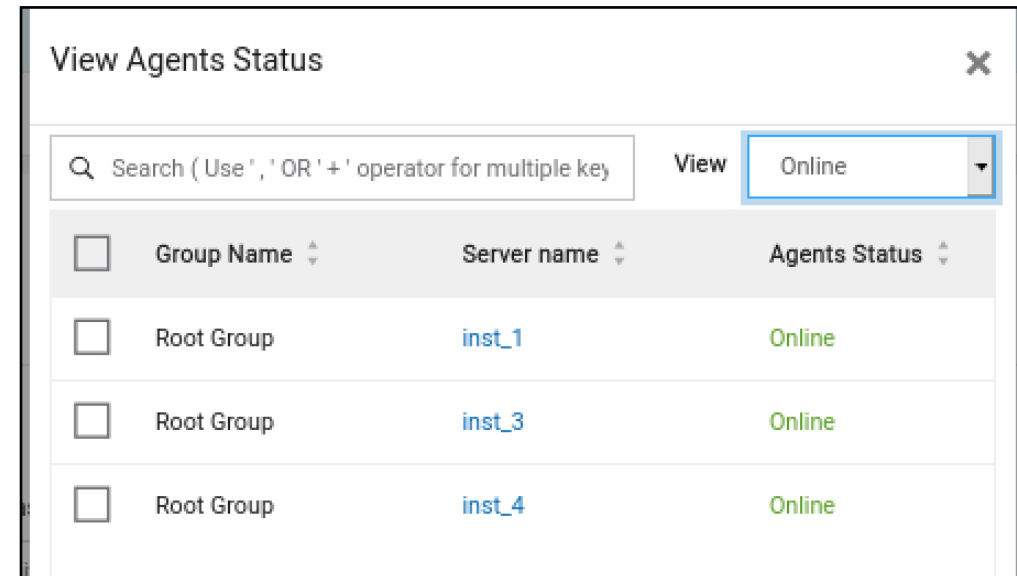
Search (Use ', ' OR ' + ' operator for multiple key) View All ▾

<input type="checkbox"/>	Group Name ▾	Server name ▾	Agents Status ▾
<input type="checkbox"/>	Root Group	inst_1	Online
<input type="checkbox"/>	Root Group	inst_2	Offline
<input type="checkbox"/>	Root Group	inst_3	Online
<input type="checkbox"/>	Root Group	inst_4	Online
<input type="checkbox"/>	Root Group	inst_5	Offline
<input type="checkbox"/>	Root Group	inst_6	Offline
<input type="checkbox"/>	Root Group	inst_7	Offline
<input type="checkbox"/>	Root Group	inst_8	Offline



View All ▾

- All
- Offline
- Online
- Not configured



View Agents Status ×

Search (Use ', ' OR ' + ' operator for multiple key) View Online ▾

<input type="checkbox"/>	Group Name ▾	Server name ▾	Agents Status ▾
<input type="checkbox"/>	Root Group	inst_1	Online
<input type="checkbox"/>	Root Group	inst_3	Online
<input type="checkbox"/>	Root Group	inst_4	Online

IHQ 2.1

- A major beneficial change occurred in the logging of errors and general information
 - Previously, Java stack traces were automatically logged creating very, very large and confusing log files
 - In IHQ 2.1, the stack traces are enabled at the default `INFO` level
 - Only the relevant and important descriptive information is logged at this level
 - The file sizes are much smaller and easier to parse
 - Users can manage to what degree and detail is logged through the `DEBUG` and `TRACE` options in the log configuration file (`server.log4j.xml` and `agent.log4j.xml`)

IHQ 2.1

- Several changes were made to the `InformixHQ.bat/sh/ksh` utility
 - First is the ability to filter the `list` output by keyword to refine the result set
 - Stopping either the server or agents is now interactive
 - Can terminate all with one command or choose which to stop
- Some user friendly interface enhancements are introduced
 - The output from the **Schema Manager:Stored Procedures & Functions** options is now in tabular format for easier reading and understanding
 - The **Storage:Logs** option which displays information about instance logical logs has more details in a user friendly view including
 - The current log
 - Log usage
 - Hovering over a graph displays the numeric percentage
 - Backup status of logs
 - The **Storage:Spaces view as chunks** window now provides the ability to copy the chunk path for any given chunk



Announcements

Announcements

- The FC8 release included a number of announcements of changes made and changes to come
- HP-UX
 - Informix stated that support for HP-UX will stop “soon”
 - This is due to the vendor’s end-of-life cycle on the product in December 2025
 - Informix will be removing support for HP-UX in future V.12 and V.14 fix packs
 - There isn’t a definite date yet when that will happen
 - It’s reasonable to assume it will occur in 2023 or early 2024
 - HP-UX will NOT be a supported port in the V.Next release scheduled for mid 2023
 - The Informix team is ready and willing to help customers needing to migrate from HP-UX to another supported platform
 - `cdr migrate server` and other migration tools are available as well as service partners

Announcements

- Removal of the embedded JRE
 - With the release of V.14.10, Informix required a Java Runtime environment at 1.8 or higher to install
 - That notwithstanding, the engine still had an internal JRE which it used for Java UDRs and other internal processes
 - In the very near future, the internal JRE (found in `$INFORMIXDIR/extend/krakatoa/jre`) will be removed
 - You will be required to set `JAVA_HOME` to the JRE you installed for installation and other Java-based processing
 - With this change, you can manage the Java level for your environment and install patches / upgrades according to your needs and requirements

Announcements

- Removal of Communication Support Modules
 - Future Informix releases will no longer contain the Encryption Communication Support Module (ENCCSM) nor the Generic Security Services Communication Support Module (GSSCSM)
 - These use older, less secure encryption protocols and methods
 - Informix moved to TLS for communication security
- Older TLS versions are now disabled
 - Transport Layer Security is the standard for securing network-based communication
 - Support for older versions, specifically 1.0 and 1.1 has been removed
 - You must use versions 1.2 (default) or 1.3

Questions

