

Introducing IBM Informix 15

Carlton Doe

Informix Technical
Product Manager

cdoe@us.ibm.com



IBM Informix 15

Introducing Informix v.15

- Informix v.15 was released in November 2024
- This release is the third largest change to Informix in its life
 - Version 9.x with the addition of object oriented data types, functionality, and flexibility
 - Version 10.x with its larger page sizes and total storage capacity
- This presentation is intended to give you an idea of what's in this release

Introducing Informix 15

- The supported platforms are:
 - Notice the Solaris and HP-UX are NOT on the list

Operating System	Version
AIX	7.2, 7.3
Linux - Intel	RHEL 8.4, 8.5, 9 SuSE SLES 15 Ubuntu 22.04 LTS – 24.04 LTS Rocky Linux 8 – 9.1 AlmaLinux 8 – 9.1 Amazon Linux 2023
Linux Power (LE)	RHEL 8.4 SuSE SLES 15 SP5
Linux IBM Z	RHEL 8.4 SuSE SLES 15 SP5
Windows	Windows 2019 - 2022
ARM AWS Graviton2	TBD
Containerized	Docker in ICR CP4D

New version numbering



New Informix versioning

- For many, many years, Informix used the following versioning format

V14.10.xyz.a_string

Where

14.10 was the major release number

x - represented the port bit depth

F - 64 bit

U - 32 bit

T - Windows

y - represented the type of release

A - Alpha

B - Beta

C - first commercial release

D and so on - significant incremental releases

z - fixpack counter

a_string

Special build identifiers on top of
the fix pack

New Informix versioning

- With V.15 we are moving to the IBM software numbering standard
 - Available here: <https://www.ibm.com/support/pages/node/318829>

Specifically a V.R.M.F. format:

V.R.M.F-a_string

Where

- V - is the major version number
- R - is the release number
- M - is the modification number
- F - is the fix pack number
- a_string* - identifies special builds

New Informix versioning

- Parsing the numbers into more detail
 - A **version** indicates a major product release with significant changes, new functionality / features
 - It represents the start of a new product lifecycle in terms of end-of-sales and end-of-support dates
 - This release is IBM Informix V.15
 - Previous format was **v.14**
 - A **release** is an interim update to the version
 - Has some new features or functionality
 - It represents the start of a new product lifecycle in terms of end-of-sales and end-of-support dates
 - This release is IBM Informix v.15.0
 - Previous format was v.14.**10**
- The combination of a V.R. represents the start of a new product lifecycle in terms of end-of-sales and end-of-support dates
 - V.15.0 can have one set of dates
 - V.15.1 has another set of dates

New Informix versioning

- Parsing the numbers into more detail
 - A **modification** (aka Mod Pack or Refresh Pack) is a minor update
 - It may have new functionality but the primary driver is to consolidate a series of **fixes** into a single package
 - Previous format was v.14.10.xCn
 - A **fix** is mitigate against defects or security vulnerabilities
 - Fixes are cumulative so x.x.x.4 will have the content of x.x.x.3 / 2 etc.
- A new mod restarts the fix counter
- Special builds are intended for short term, emergency defect fixes are a bridge to a fix or mod
 - Identified with a text string 15.0.0.0-X#
 - A special build is supported for 90 days after an encapsulating fix or mod is released
 - Special builds do not include defects corrected and released in other fixes / mods

New Informix versioning

- The CSDK and JDBC drivers are following the same numbering format starting with this release
 - The CSDK released with V.15 is CSDK v.15.0
 - The JDBC driver released with V.15 is JDBC v.15.0

Updated Java and GSKit information



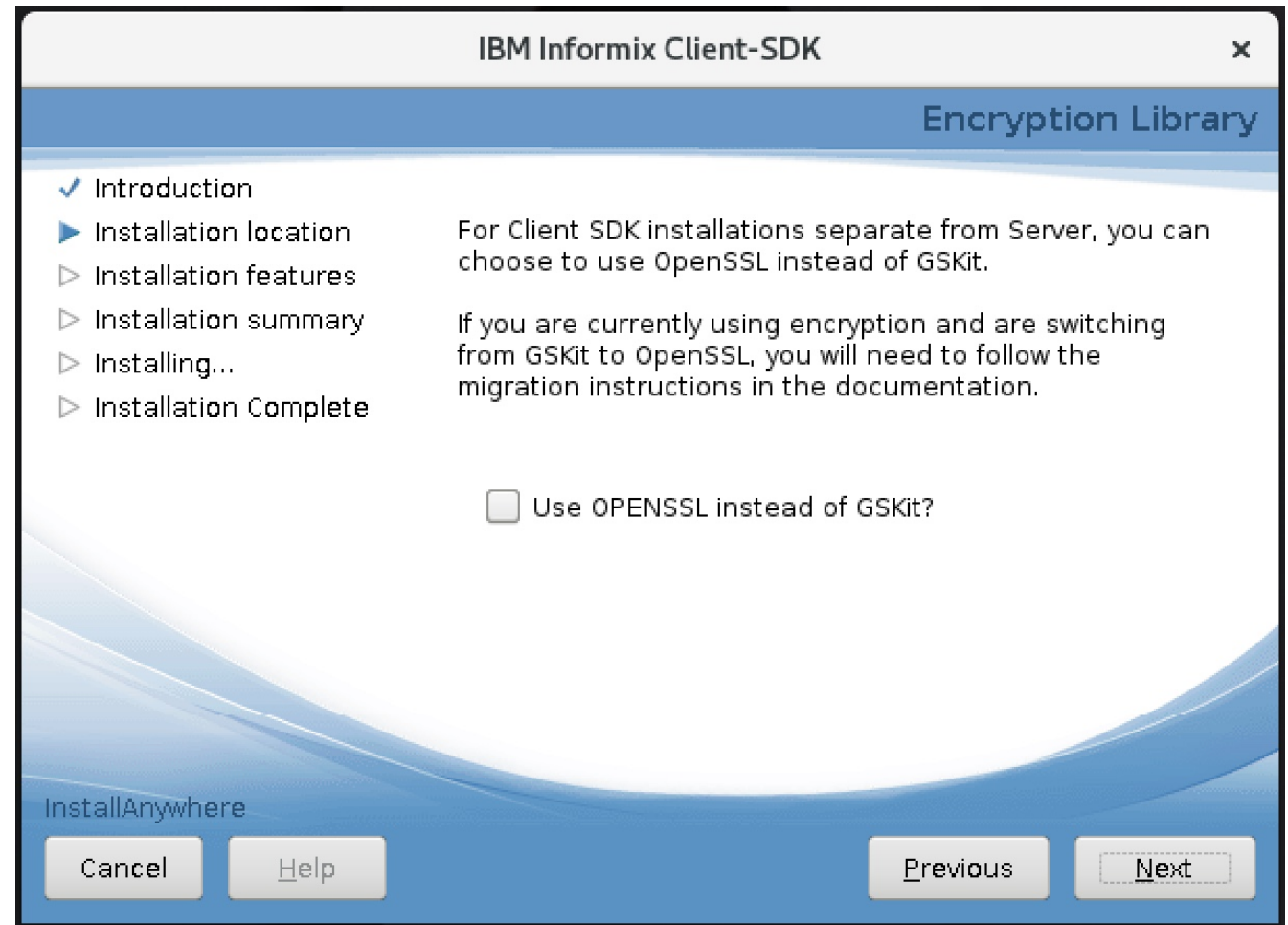
Updated Java requirements

- The same installation methods exist for v.15 as earlier versions
- Java 11 or higher is required
 - Java 11 is the minimum supported level for IHQ v.3
- For the engine / CSDK GUI install to work, ensure you don't install just the Java headless version
 - Headless can be available but also the full JRE

```
Inst1_15: rpm -qa |grep java
tzdata-java-2024a-1.el8.noarch
javapackages-filesystem-5.3.0-2.module+el8+2598+06babf2e.noarch
java-1.8.0-openjdk-headless-1.8.0.412.b08-2.el8.x86_64
java-11-openjdk-11.0.23.0.9-3.el8.x86_64 ←
java-11-openjdk-headless-11.0.23.0.9-3.el8.x86_64 ←
Inst1_15:
```

CSDK install

- For the CSDK, you have the same installation choices as in v.14.10 — whether to use GSKit or OpenSSL for client only installs
 - The engine only uses GSKit
- See next screen for more clarity



CSDK install

- A little more detail on this
 - When installing the CSDK on the same server / installation location as the engine, the GSKit is installed
 - With clients, you have the choice of either library (GSKit or OpenSSL for the SSL protocol)
 - We do not install OpenSSL (except on Windows), but configure the product to use it
 - If you are using a '.p12' keystore on clients, OpenSSL and GSKIT support the same .p12 keystore so you don't even need different keystores to support OpenSSL
 - You do, however, need to update the stash file via `onkstash`
 - If you are using the GSKit's ".kdb" keystore format on the client, you will need follow the migration instructions in the documentation to convert ".p12" keystores before you can use OpenSSL
 - For example:

```
gsk8capicmd_64 -keydb -convert -db <keystore.kdb> -new_db <keystore.p12> -stashed  
-pqc false
```

CSDK install

- Further explanation from development

*The part that needs clarity is that both GSKit and OpenSSL are **libraries** used to support the SSL protocol for encrypted communication between clients and servers. The certificates and keystores are the same for both.*

The only issue is that because of the proprietary nature of the GSKit password stash algorithm, we can't use it for OpenSSL. There is no issue of using OpenSSL for clients and GSKit for servers. AFAIK, nothing has changed here for V15.

Changes to \$ONCONFIG



`$ONCONFIG` changes

- There are changes in the `$ONCONFIG` file
 - Some parameters were removed
 - A few new ones were added
 - A few have new default values
 - Because of potential new functionality, some parameters have more of an impact on instance processing
 - We have updated their description to explain how they work in the new environment
- If you're not familiar with it, the `onconfig_diff` utility can be used to compare two `onconfig` files to highlight differences
 - We encourage you to look carefully at the new `onconfig` file, parameter defaults and new parameters in particular when considering migration to V.15
 - Some of the new defaults or parameters may be critical to proper instance operation

\$ONCONFIG changes

- For example:

```
Ifmx-IWA: onconfig_diff -c -f ../etc/onconfig.std -s ../../14_10/etc/onconfig.std
```

```
=====
File 1: ../etc/onconfig.std
File 2: ../../14_10/etc/onconfig.std
=====
```

```
Parameters Found in File 1, not in File 2
=====
```

```
BAR_GROUP
    bargroup

BAR_BSALIB_PATH
    $INFORMIXDIR/lib/libbsapsm.so

PSM_CATALOG_PATH
    $INFORMIXDIR/etc/psm

VPCLASS
    jvp,num=0

JVPJAVAHOME
    $INFORMIXDIR/jre
```

```
=====
Parameters Found in File 2, not in File 1
=====
```

```
CDR_MEM
    0
```

```
=====
Parameters Found in both files, but different
=====
```

```
PHYSBUFF
```

```
File 1: 4096
File 2: 128
```

```
LOGBUFF
```

```
File 1: 128
File 2: 64
```

```
BAR_XFER_BUF_SIZE
```

```
File 1: 128
File 2: 31
```

```
SEC_LOGREC_MAXBUFS
```

```
File 1: 0
File 2: 1000
```

```
BUFFERPOOL
```

```
File 1: default,buffers=10000,lrus=8,lru_min_dirty=50,lru_max_dirty=60
File 2: default,buffers=10000,lrus=8,lru_min_dirty=50.00,lru_max_dirty=60.50
```

```
Ifmx-IWA: █
```

The big Kahuna literally!

aka storage scalability



Storage scalability

- The central theme of v.15.0 is virtually unlimited scalability
 - Some customers have run into capacity limits that caused the instance to fail
 - Most of the time due to their lack of instance monitoring 🤔
- The goal of this release was to eliminate those limits for all intents and purposes
 - You have a “big data” problem? It is trivial to solve with Informix
- There’s big and then there’s **big** so how big are we talking about??

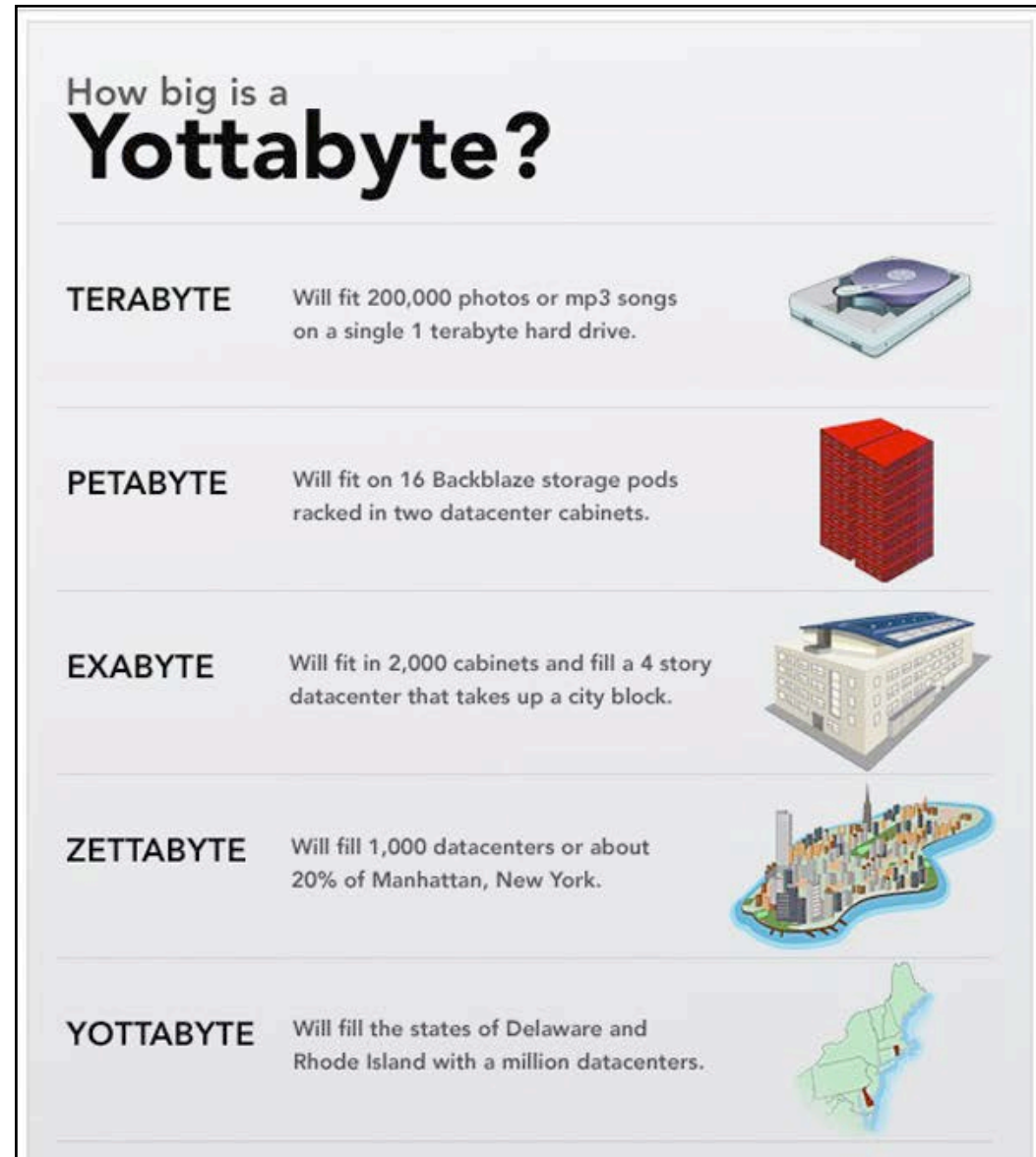
1/2 yottabyte !!!!!

Storage scalability

- What's a yottabyte??
 - It's the largest unit approved by the International System of Units (SI)
 - The yottabyte is about 1 septillion bytes -- or, as an integer, 1,000,000,000,000,000,000,000,000 bytes
 - The storage volume is equivalent to a quadrillion gigabytes (GB) or a million trillion megabytes
- It would cost ~\$50 trillion in storage to max out the engine!



Are you happy now?



Storage scalability

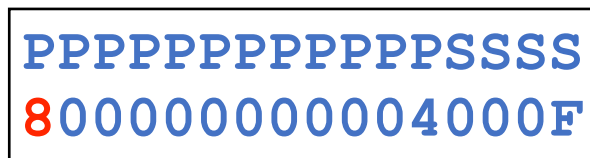
- How do we get there?
 - Row size remains unchanged at 32 kb
 - This will probably change in the future
 - RowIDs change from 4 bytes to 8 bytes
 - The engine supports legacy (Infrastructure Version 0) and new rowIDs (Infrastructure Version 1)
 - For backward and forward compatibility
 - Internally the engine uses 8 byte containers for all rowIDs

- “Small” tables continue to use the legacy, 4-byte rowID



Page 0x400, slot 0x0f

- “Large” tables use the larger 8 byte rowIDs
 - The sign bit is set to distinguish its value from a 4 byte rowID



Page 0x4, slot 0x000f

Storage scalability

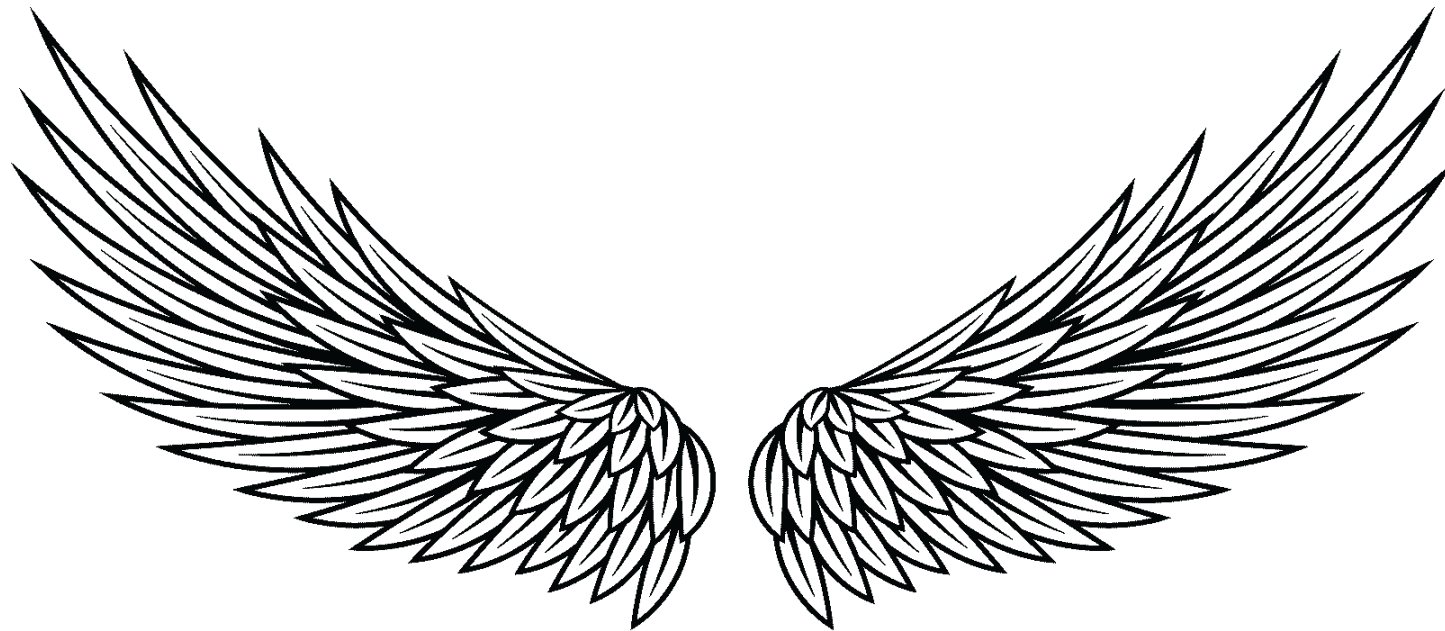
- How did we get there?
 - Page size
 - Depending on the port default page size, page size options have been
 - 2k - 2, 4, 6, 8, 10, 12, 14, 16 kbs
 - 4k - 4, 8, 12, 16 kbs
 - With V.15.0, additional page sizes are available for both default page sizes
 - For example - 32, 64, 128, 256 kbs
 - Slots / data page - # of rows / page
 - Historically this was limited to 255, the new effective limit is 30,000
 - Based on what we're trying to do, the theoretical maximum is 65,534 but don't push it!
 - Page address changes
 - Page offsets change from 4 bytes to 8 bytes
 - The 2 byte chunk number becomes an unsigned `smallint`

Storage scalability

- How did we get there?
 - The max pages / RSAM partitions values have increased
 - From 16,777,215 to 140,737,488,355,327
 - The max chunk size was increased as well
 - The current limit is 4 TB but in V.15 the new limit is 9 PB!!!
 - This means that one V.15 chunk can be larger than the current storage limit of V.12.10 / V.14.10!!
 - In theory we could have even bigger chunks but we are limited by O/S `lseek()` functionality
 - 9 PB is as big of an object most O/Ss can read / write to
 - The maximum number of chunks was increased as well
 - From 32,767 to 65,534

Storage scalability

- How did we get there?
 - The maximum index key size was expanded
 - From the current 3kb to as large as 32kb (on pages > 32kb)
 - Do you really need an index that's 32kbs??
 - Page timestamps were altered as well
 - Changing from 4 bytes to 8 bytes
 - This has implications for incremental backups
 - With the 4 byte length, the timestamps rolled over too quickly to be useful
 - Incremental backups don't work properly
 - BTW, how quick was "quick?" - it was machine dependent (CPU speed and database load are factors)
 - Best case, maybe a day stretching into a second day to worst case, just a couple of hours
 - With the longer page timestamp length, it should take 50 years for the timestamps to wrap and incremental backups now function as intended



Shall we stretch our wings??

Larger page sizes and more rows per page



Larger page sizes and more rows per page

- You can use the new page sizes in conjunction with legacy page sizes
 - In IV-1
- For example

```
Dbspaces
address      number  flags      fchunk  nchunks  pgsize  flags      owner    name
46164028    1       0x4030001  1        1         2048    N BA      informix rootdbs
47b10cb8    2       0x4020001  2        1         262144  N BA      informix data_space_1
48966a68    3       0x4020001  3        1         262144  N BA      informix data_space_2
47b10708    4       0x4030001  4        1         2048    N BA      informix log_space
47d17a68    5       0x4028001  5        1         2048    N SBA     informix slob_space
47d253b0    6       0x4002001  6        1         262144  N TBA     informix work_space
4805ea68    7       0x400a001  7        1         2048    N UBA     informix slob_space_temp
4815dc30    8       0x4020001  8        1         262144  N BA      informix data_space_3
8 active, 2047 maximum
```

Larger page sizes and more rows per page

- As you might expect, buffer pools of matching sizes to the new page sizes are available

```
-----  
Buffer pool page size: 262144  
dskreads  pagreads  bufreads  %cached  dskwrits  pagwrits  bufwrits  %cached  
19845     2544640  191389   89.63    9          1152      9          0.00  
bufwrits_sinceckpt  bufwaits  ovbuff  flushes  
0           3         0       2  
  
Fg Writes      LRU Writes      Avg. LRU Time  Chunk Writes  Total Mem  
0              0              -nan           9             2502Mb  
  
Bufferpool Segments  
id segment      size      # buffs  
0 0x65991000    2502Mb   10000  
  
-----  
  
Fast Cache Stats  
gets      hits      %hits    puts  
86092     83173    96.61    73433
```

Larger page sizes and more rows per page

- If some is good, is “more” always better??
 - Perhaps not
 - Yes, you can pack more data into a single 256 k page than with 128 x 2k pages
 - Saving 127 sets of headers and timestamps out of the total space used
 - Yes, I/O overhead will decrease since more data is being returned in a single operation
 - But with very small row or index entry lengths you may have more slot table overhead per row on a “large” page db space than with smaller pages, thus wasting space & reducing row count
 - A potential example - 10 byte row length

On a 2 k page size

```

Headers:                32 * 128
Page-ending timestamps: 8 * 128
Slot table entries:     4 * 143 * 128

Total overhead: 78336 bytes
Unused bytes per page: 2048 - (32+8) - ((10 + 4) *
143) = 6 bytes per page
Number of rows packed into 256 kilobytes: 18304
  
```

On a 256 k page size

```

Headers:                32 * 1
Page-ending timestamps: 8 * 1
Slot table entries:     8 * 14561 * 1

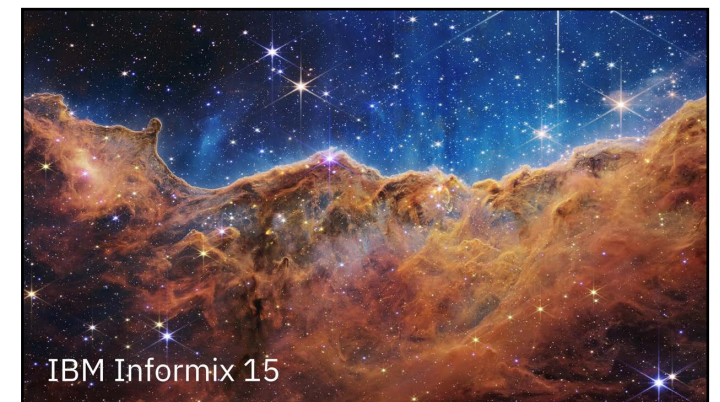
Total overhead: 116528 bytes
Unused bytes per page: 262144 - (32+8) - ((10 + 8) *
14561) = 6 bytes per page
Number of rows packed into 256 kilobytes: 14561
  
```

Larger page sizes and more rows per page

- Some guidance from development for those contemplating using larger page sizes in IV-1

The bottom line is that the choice of page size [for tables and indexes] should be made after some testing and benchmarking. It's hard right now to list all the pros and cons--I don't think we'll know for a bit.

Larger chunk size and number of pages per partition



Larger chunk size / number of pages

- As already mentioned, we are limited by the `lseek()` function to the theoretical maximum on chunk sizes
 - That notwithstanding, maximum chunk size was significantly expanded
 - For example, an 11 TB chunk / dbspace (using 2 k page size)

Chunks	address	chunk/dbs	offset	size	free	bpages	flags	pathname
	460ec2a8	1 1	0	150000	82338		PO-B--	/work1/
	483db028	2 2	0	5905580031	5888800818		PO-BE-	/bigfs/
	483dc028	3 3	0	625	56		PO-BE-	/work1/
	483dd028	4 4	0	524288	524235		PO-BE-	/work2/
4 active, 65534 maximum								

Larger chunk size / number of pages

- Similarly, many customers have run into a limitation of ~16 million pages / partition and want / need more
 - You can now have over *140 trillion!!*
- A test load using over 33 million pages

		TBLspace uses large rowids
Maximum row size	1504	
Number of special columns	0	
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	16777215	
Next extent size	16777215	
Number of pages allocated	33554430	
Number of pages used	32955903	
Number of data pages	32947664	
Number of rows	32947664	
Partition partnum	2097251	
Partition lockid	2097251	
Last DML time	No Record	
First V1 Bitmap Page	0	

Index size



Index size

- In Informix v.12.10 / 14.10, index size is limited to ~3K when created on a 16k page size
- In conjunction with larger page sizes, the maximum index size limit was expanded to ~32k on a 256 k page size
 - Unlike data pages which can have home and remainder pages, index entries must fit on a single page
 - The creation of indexes in different page size dbspaces should be reviewed / calculated to avoid as much wasted space as possible

Index size

- For example, trying to create an index on a ~28k column fails on a 16k page dbspace but succeeds on a 256k page dbspace

```
drop table if exists lrg_idx_tab;
create table if not exists lrg_idx_tab
( col1 smallint,
  col2 char(28000))
in data_space_4;

insert into lrg_idx_tab values (1, "some text");
insert into lrg_idx_tab values (2, "more text");

create index my_lrg_idx on lrg_idx_tab (col2) in data_space_4;
#
# 517: The total size of the index is too large or too many parts in index.
#
```

pgsize	flags	owner	name
2048	N BA	informix	rootdbs
262144	N BA	informix	data_space_1
262144	N BA	informix	data_space_2
2048	N BA	informix	log_space
2048	N SBA	informix	slob_space
262144	N TBA	informix	work_space
2048	N UBA	informix	slob_space_temp
262144	N BA	informix	data_space_3
16384	N BA	informix	data_space_4

```
drop table if exists lrg_idx_tab;
create table if not exists lrg_idx_tab
( col1 smallint,
  col2 char(28000))
in data_space_2;

insert into lrg_idx_tab values (1, "some text");
insert into lrg_idx_tab values (2, "more text");

create index my_lrg_idx on lrg_idx_tab (col2) in data_space_2;

index created
```

How do you get to these
larger limits?
aka Infrastructure Version 1



Getting to Infrastructure Version 1

- Similar to Informix v.10 where there was a “legacy” and “new” format to sizing and limits, v.15 has the same distinction
 - The legacy format is called “Infrastructure Version 0” (IV-0) and the new format is called “Infrastructure Version 1” (IV-1)
- Migration from legacy (IV-0) to IV-1 environments depends on several factors
 - Legacy (migrated) instances are in IV-0 until a migration to IV-1 command is issued
 - Newly initialized instances are created in IV-1
- Similar to V.10, IV-0 instances can be rolled back to earlier Informix versions such as v.14.10 if necessary
 - Once the instance is migrated to IV-1, version rollback will not be possible
 - Only a full restore from backup

Getting to Infrastructure Version 1

- There are some table elements from legacy sized tables that will be different in an IV-1 environment
 - These will be discussed in greater detail in my technical deep dives
- Even if you're in an IV-1 environment, keywords and commands are available that impose some IV-0 object limits for application compatibility
 - These will be discussed in greater detail in my technical deep dives
- Some instance objects remain unchanged regardless of the infrastructure version such as
 - Tables in the system databases (`sysmaster`, `sysadmin`, `syscdr`, `sysuser`, `sysutils`)
 - The `Tblspace` `Tblspace`
 - Internal SLOB space partitions
 - System catalog tables (`sysables`, `syscolumns` etc.)
 - The `database` `tablespace`

Getting to Infrastructure Version 1

- I will provide much more detail and guidance on how to use these larger limits
 - Look for roadshows or other enablement events

- We are releasing a new fix pack to 4GL and ISQL to take advantage of the new table and other size functionality added to the database engine
 - These products retain their current version numbering scheme

Migrating from earlier versions or with replication



Migration

- There are SIGNIFICANT changes to the Informix v.15 kernel from earlier versions
- When migrating an existing instance to v.15, a specific process must be followed
 - After migration, the instance is in a “compatibility mode”, aka Infrastructure Version 0 (IV-0)
 - You can process work and if necessary, use the builtin reversion utility to roll back to the earlier Informix version
- Once satisfied with the instance, you can trigger the full v.15 functionality at which point reversion is not possible
 - You must restore from an earlier version backup

Migration

- To migrate an existing instance to v.15
 - All in-place alters must be processed and removed
 - While highly recommended for earlier version changes, it will be required in v.15
 - If not processed, the conversion will fail with messages indicating IPAs exist
 - No conversion tasks are executed
 - You must shut the instance down and restart using the earlier Informix version
 - Process the IPAs then try the upgrade again

Migration

- You can check for tables with outstanding in-place alters by
 - Enabling the `check_for_ipa` task in the database Scheduler
 - Any table with outstanding IPAs is listed the `ph_alert` table
 - Execute an `oncheck -pT` on all tables looking for multiple “versions” in the **Home Data Page Version Summary** section

Home Data Page Version Summary	
Version	Count
0 (oldest)	2
1	3
2	0
3	26
4 (current)	3

Getting to Infrastructure Version 1

- Since Informix gives you choices, you have 5 different workflows to get to Infrastructure Version 1

- In all cases, make sure you have a level 0 backup before proceeding
 - Remember no in-place reversion is possible once moved to IV-1

1. `ontape -s -L 0 -IV1`

2. `onmode -wf TAPEDEV=/dev/null`
`ontape -s -L 0 -IV1` } (Creates a fake archive then sets the IV-1 flag)

3. `onbar -b -w -IV1` (The full system flag (`-w`) must be set)

4. `onbar -b -F -IV1`

5. Through the `sysadmin` API

`execute function [task | admin] ('archive fake IV1');`

(These two operations create a fake archive then sets the IV-1 flag)

Migration

- I will provide significantly more details on what the migration process looks like, including error handling in my technical deep dives
- For those with replication enabled, depending on which technology you are using, the migration process will vary
 - For example, those using ER can migrate following a suggested methodology and keep your ER configuration
 - Since the `sysmaster` database is rebuilt as part of the v.15 upgrade, those using the H/A cluster must rebuild the cluster
 - You can not boot into physical recovery mode then engage the logical recovery process
 - Look for more details in my technical deep dives

New GLS libraries



New GLS libraries

- With this release, Informix has moved to ICU4c 71.1
 - What changed?
 - ICU 71 adds phrase-based line breaking for Japanese
 - Existing line breaking methods follow standards and conventions for body text but do not work well for short Japanese text, such as in titles and headings. This new feature is optimized for these use cases.
 - ICU 71 adds support for Hindi written in Latin letters (`hi_Latn`)
 - ICU 71 updates to the time zone data version 2022a
 - Pre-1970 data for a number of time zones has been removed
 - The `DateTimePatternGenerator` now uses the appropriate date-time combining pattern as specified by `CLDR` for skeletons which combine date and time elements
 - For example: Thursday, February 22, 12:30 PM to Thursday, February 22 **at** 12:30 PM
 - More

Getting the query plan of a running query



SQL enhancements - running query plan

- We added two options to get the query plan of a running query
 - An additional `onstat` flag
 - Yeah!! Another flag for the utility that keeps on giving
 - An addition column to the `syssexplain` pseudo table
- With the `onstat` command, the full query plan is returned
 - If the query plan exceeds 32k, the plan output is truncated in the `syssexplain` pseudo table
- Both operations do not contain runtime statistics like those shown in `sql explain`
 - The query is still running so it's not possible to have a complete set of values

SQL enhancements - running query plan

- Using the `onstat -g qplan [sess_id | 0]` flag returns information for either a specific session or all active sessions
 - Example, a session in a DW environment
 - Session ID: 40

```

Inst_1: onstat -u
Your evaluation license will expire on 2024-03-20 12:41:39

IBM Informix Dynamic Server Version 15.0.FC0TL -- On-Line -- Up 00:09:00 -- 3113428 Kbytes
2024-01-09 17:06:54 -- Infrastructure Version: 1

Userthreads
address      flags  sessid  user    tty    wait          tout locks nreads nwrites
46322028    ---P--D 1      informix -      0          0      0    14443 2570
46322af0    ---P--F 0      informix -      0          0      0      0    3474
463235b8    ---P--F 0      informix -      0          0      0      0    258
46324080    ---P--F 0      informix -      0          0      0      0    656
46324b48    ---P--F 0      informix -      0          0      0      0      0
46325610    ---P--F 0      informix -      0          0      0      0      0
463260d8    ---P--F 0      informix -      0          0      0      0      0
46326ba0    ---P--F 0      informix -      0          0      0      0      0
46327668    ---P--F 0      informix -      0          0      0      0      0
46328130    ---P--- 5      informix -      0          0      0      0      0
46328bf8    ---P--B 6      informix -      0          0      0      0      0
463296c0    Y--P--D 7      informix -    46f5ef00    0      0    982496 0
4632a188    ---P--D 8      informix -      0          0      0      0      0
4632ac50    ----- 1      informix -      0          0      0      0      0
4632b718    ---P--D 22     informix -      0          0      0      0      0
4632c1e0    Y--P--D 42     informix -    441a8048    0      0      0      0
4632ed00    Y--P--- 40     ihqmonit 2    4e79ed18    0      1    129      0
4632f718    P----- 38     informix -      0          0      0      0      0
  
```

SQL enhancements - running

- Running the `onstat` command
 - Estimated costs are included

```

Inst_1: onstat -g qplan 40
Your evaluation license will expire on 2024-03-20 12:41:39

IBM Informix Dynamic Server Version 15.0.FC0TL -- On-Line -- Up 00:15:02 -- 3113428 Kbytes
2024-01-09 17:12:57 -- Infrastructure Version: 1

Session [40]
QUERY:
-----
select first 100 sum(ws_net_paid) as total_sum, i_category, i_class
  from
    web_sales, date_dim d1, item
  where
    d1.d_year = 2002
    and d1.d_date_sk = ws_sold_date_sk
    and i_item_sk = ws_item_sk
  group by 2,3
  order by 2,3

QUERY: (OPTIMIZATION TIMESTAMP: 01-09-2024 17:12:57)Estimated Cost: 324043
Estimated # of Rows Returned: 110
Maximum Threads: 8
Temporary Files Required For: Order By  Group By

  1) ihqmonitor.d1: SEQUENTIAL SCAN

      Filters: ihqmonitor.d1.d_year = 2002

  2) informix.web_sales: INDEX PATH

      (1) Index Name: informix.ind_121
          Index Keys: ws_sold_date_sk  (Parallel, fragments: ALL)
          Lower Index Filter: ihqmonitor.d1.d_date_sk = informix.web_sales.ws_sold_date_sk
NESTED LOOP JOIN

  3) informix.item: INDEX PATH

      (1) Index Name: informix.ind_10
          Index Keys: i_item_sk  (Parallel, fragments: ALL)
          Lower Index Filter: informix.item.i_item_sk = informix.web_sales.ws_item_sk
NESTED LOOP JOIN

```

SQL enhancements

- Using 0 (zero) with the `onstat` command returns the query plan for all active instance sessions, including system related

```

Inst_1: onstat -g qplan
Your evaluation license will expire on 2024-03-20 12:41:39

IBM Informix Dynamic Server Version 15.0.FC0TL -- On-Line -- Up 00:04:51 -- 3113428 Kbytes
2024-01-11 10:27:49 -- Infrastructure Version: 1

Session [45]: No running query to generate the plan. ←
Session [44] ←
QUERY:
-----
select first 100 sum(ws_net_paid) as total_sum, i_category, i_class
  from
    web_sales, date_dim d1, item
  where
    d1.d_year = 2002
    and d1.d_date_sk = ws_sold_date_sk
    and i_item_sk = ws_item_sk
  group by 2,3
  order by 2,3

QUERY: (OPTIMIZATION TIMESTAMP: 01-11-2024 10:27:50)Estimated Cost: 8335091
Estimated # of Rows Returned: 19380
Temporary Files Required For: Order By Group By

1) informix.web_sales: SEQUENTIAL SCAN

2) ihqmonitor.d1: AUTOINDEX PATH

   Filters:
   Table Scan Filters: ihqmonitor.d1.d_year = 2002

   (1) Index Name: (Auto Index)
       Index Keys: d_date_sk
       Lower Index Filter: ihqmonitor.d1.d_date_sk = informix.web_s
NESTED LOOP JOIN

3) informix.item: AUTOINDEX PATH

   (1) Index Name: (Auto Index)
       Index Keys: i_item_sk
       Lower Index Filter: informix.item.i_item_sk = informix.web_sales.ws_item_sk
NESTED LOOP JOIN

```

```

Session [38]: No running query to generate the plan.
Session [37]: No running query to generate the plan.
Session [36]: No running query to generate the plan.
Session [35]: No running query to generate the plan.
Session [3]: No running query to generate the plan.
Session [2]: No running query to generate the plan.

```

SQL enhancements - running query plan

- The `sqx_sqlstatementplan` column was added to the `syssexplain` pseudo table in the `sysmaster` db
 - Like the `onstat` command, it only works for active sessions
 - There isn't any history of past sessions and query plans
- Example
 - Using the same session from the previous slide

```
----- sysmaster@inst_1 ----- Press  
select * from syssexplain where sqx_sessionid = 47
```

SQL enhancements - running query plan

```
sqx_sessionid      47
sqx_sdbno          0
sqx_iscurrent      Y
sqx_executions     0
sqx_cumtime        0.00
sqx_bufreads       0
sqx_pagereads      0
sqx_bufwrites      0
sqx_pagewrites     0
sqx_totsorts       0
sqx_dsksorts       0
sqx_sortspmax      -1
sqx_conbno         0
sqx_ismain         Y
sqx_selflag        SQ_SELECT
sqx_estcost         9723407
sqx_estrows        8333
sqx_seqscan        -1
sqx_srtscan        -1
sqx_autoindex      -1
sqx_index          -1
sqx_remsql         -1
sqx_mrgjoin        -1
sqx_dynhashjoin    -1
sqx_keyonly        -1
sqx_tempfile       -1
sqx_tempview       -1
sqx_secthread      -1
sqx_sqlstatement   select first 1000 a.ca_state state, count(*) cnt
                    from
                    customer_address a, customer c, web_sales s, date_dim
                    d, item i
                    where a.ca_address_sk = c.c_current_addr_sk
                    and c.c_customer_sk = s.ws_bill_customer_sk
                    and s.ws_sold_date_sk = d.d_date_sk
                    and s.ws_item_sk = i.i_item_sk
                    and d.d_month_seq >= 2
                    and i.i_current_price > 50.00
                    group by a.ca_state
                    having count(*) >= 10
                    order by cnt
```

sqx_sqlstatementp+

```
QUERY: (OPTIMIZATION TIMESTAMP: 01-09-2024 17:21:28)Estimated
Cost: 9723407
Estimated # of Rows Returned: 8333
Maximum Threads: 10
Temporary Files Required For: Order By Group By

1) ihqmonitor.d: SEQUENTIAL SCAN

Filters: ihqmonitor.d.d_month_seq >= 2

2) ihqmonitor.s: INDEX PATH

(1) Index Name: informix.ind_121
Index Keys: ws_sold_date_sk (Parallel, fragments:
ALL)
Lower Index Filter: ihqmonitor.s.ws_sold_date_sk =
ihqmonitor.d.d_date_sk
NESTED LOOP JOIN

3) ihqmonitor.i: INDEX PATH

Filters: ihqmonitor.i.i_current_price > 50

(1) Index Name: informix.ind_10
Index Keys: i_item_sk (Parallel, fragments: ALL)
Lower Index Filter: ihqmonitor.s.ws_item_sk = ihqmo
nitor.i.i_item_sk
NESTED LOOP JOIN

4) ihqmonitor.c: INDEX PATH

(1) Index Name: informix.ind_13
Index Keys: c_customer_sk (Parallel, fragments: A
LL)
Lower Index Filter: ihqmonitor.c.c_customer_sk = ih
qmonitor.s.ws_bill_customer_sk
NESTED LOOP JOIN

5) ihqmonitor.a: INDEX PATH

(1) Index Name: informix.ind_2
Index Keys: ca_address_sk (Parallel, fragments: A
LL)
Lower Index Filter: ihqmonitor.a.ca_address_sk = ih
qmonitor.c.c_current_addr_sk
NESTED LOOP JOIN
```

SQL enhancements - running query plan

- In the previous example, there are a number of counters which are not populated
 - More guidance will be given in my technical deep dives

“Invisible” index



Invisible index

- There are times when debugging and other testing when it would be nice to “hide” an index from operations
 - Ideally without affecting other operations that are still running
- There are three existing workflows to accomplish this
 1. An explicit directive - comments included in the SQL statement telling the optimizer how to build the query plan
 - “Do this” or “don’t do this” types of directions
 - Some keywords used - `avoid_multi_index`, `full`, `index`, `use_nl`, `avoid_hash` etc.
 2. An external directive - directions as well as the SQL statement are stored in the `sysdirectives` table
 - Only available if the `IFX_EXTDIRECTIVES $ONCONFIG` and client-side parameter are enabled
 - Values: 0 (off), 1 (directive is active if the client sets the parameter), 2 (directives always used)
 - When a matching SQL statement is executed, those directions are applied to the operation
 - This is expensive since a catalog match is executed for each operation

Invisible index

- There are three existing workflows to hide an index
 3. Disabling the index through the `set indexes [idx_name | for tab_name] disabled` command
 - This affects all users of the disabled index(es)
 - Any potential changes to the index(es) due to DML operations are not written to the index
 - The index must be completely rebuilt when re-enabled

Invisible index

- With V.15, a new syntax flag was added to make an end-user index “invisible” when creating or using the index
 - `resource` privileges on the database are required to execute the command
 - For example:
 - Index creation

```
create index idx_name on tab_name (columns) invisible;
```

```
create index idx_name on tab_name (columns) visible; (default action)
```
 - In an SQL statement

```
set indexes idx_name invisible;
```

```
set indexes idx_name visible;
```

```
set indexes for tab_name invisible;
```

```
set indexes for tab_name visible;
```
- System generated indexes (e.g. PK, FK from direct creation etc.) are NOT affected by these operations
 - See the V.14.10 documentation or my 12.10.FC8 and 14.10 presentations on converting / renaming system generated indexes or constraints to end-user objects

Invisible index

- Unlike a disabled index, an invisible index is “maintained” so DML operations changes are reflected in the index contents
 - You don’t need to rebuild the index when it’s made “visible” again

Invisible index

- You can change state (visible to invisible and vice versa) while sessions are actively accessing the table referenced in the command
 - Potential example
 - `ind_122` on the `web_sales` table

```
----- ds2@inst_1 --  
set indexes ind_122 invisible  
  
Mode set.
```

Invisible index

- A `dbschema` of the table with the `-ss` flag shows the updated index creation command

```

create index "informix".ind_120 on "informix".web_sales (ws_ship_mod
  using btree in data_space_3;
create index "informix".ind_121 on "informix".web_sales (ws_sold_dat
  using btree in data_space_3;
create index "informix".ind_122 on "informix".web_sales (ws_sold_tim
  using btree in data_space_3 invisible ; ←
create index "informix".ind_123 on "informix".web_sales (ws_warehouse
  using btree in data_space_3;
create index "informix".ind_124 on "informix".web_sales (ws_web_page

```

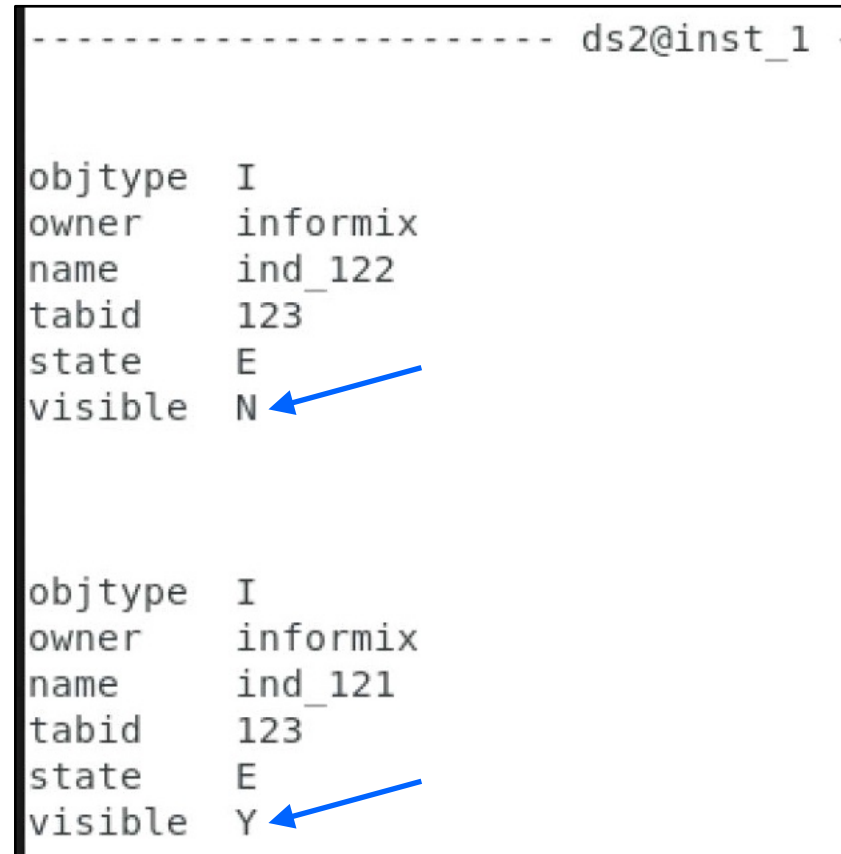
idxname	ind_122
owner	informix
tabid	123
idxtype	D
clustered	
levels	2
leaves	549.0000000000
nunique	84502.00000000
clust	486800.00000000
nrows	7200681.000000
indexkeys	2 [1]
amid	1
amparam	
collation	en_US.819
pagesize	262144
nhashcols	0
nbuckets	
ustlowts	2024-01-10 09:48:54.00000
ustbuildduration	0:00:00.00000
nupdates	0.00
n deletes	0.00
ninserts	0.00
fextsize	0
nextsize	0
indexattr	16 ←
jparam	

- Looking at the `sysindices` (not `sysindexes`) table
 - If `indexattr = 16`, that index is invisible
 - If `indexattr =`, that index is visible

Invisible index

- Another option to see the index state is to query the `sysobjstate` table which has a new `visible` column
 - A “-“ in the column indicates visibility is not an option for that object, such as constraints, triggers etc.

```
----- ds2@inst_1 -----  
  
objtype  I  
owner    informix  
name     ind_122  
tabid    123  
state    E  
visible  N  
  
objtype  I  
owner    informix  
name     ind_121  
tabid    123  
state    E  
visible  Y
```



Invisible index

- I will provide more details on this feature in my technical deep dives

IHQ



IHQ

- With Informix v.15, IHQ was upgraded to v.3.0.0
 - If you are currently using a version released prior to V.14.10.FC8, you MUST follow the upgrade process described in the 14.10.FC8 deep dive webcast to migrate the H2 database to the new version
 - Otherwise, you'll corrupt your IHQ environment and probably have to rebuild from scratch
 - This release requires Java 11 (minimum) or higher to function
 - You are required to have the appropriate JRE to support the tool
- Some of the new features include
 - Infrastructure changes
 - Light modes
 - Offline functionality
 - Enterprise Replication, point and click cluster building
 - SQL Editor enhancements

IHQ

- IHQ v.3.0 features will be explained in greater detail in my technical deep dives

Questions

