Informix Auditing

xDB Systems

Mike Walker
mike@xdbsystems.com

14 July, 2022

Informix Auditing – What is it?

- Makes an entry in a text file and/or syslog **for events that you specify**
- Can record that there have been changes to data, schema changes and permission changes
- Can record database accesses
- Can record administrator actions (e.g. onparams, onstat, etc)
- Can track data changes/accesses to ALL tables or SELECTED tables
- Expects a DBSO/AAO role, which can be the “informix” user
- Already included in your install – no additional cost
- May satisfy an auditor requirement
- Simple to set up

Informix Auditing – What it is NOT

- Does not record SQL
- Only tracks what you have told it to track
- Does not tell you the *details* of what changed
- Does not provide automatic alerts
- It is not hands off – need to archive files, etc.
- Works at the *instance* level – not the database level
- Requires role separation for improved security (who has the *informix* password?)

Audit File

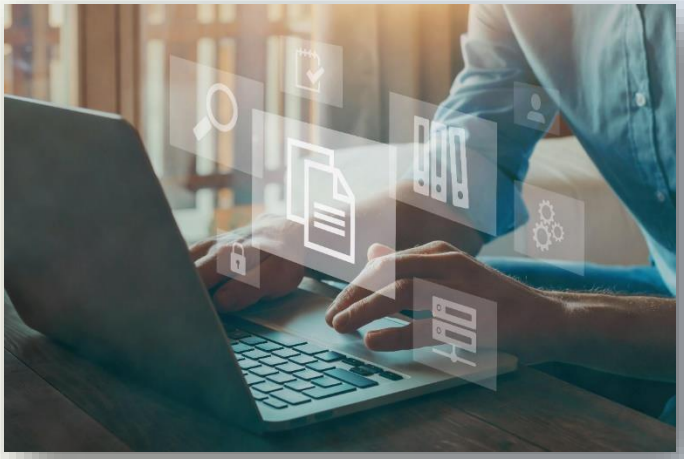
```
ONLN|2022-06-29
22:23:10.000|localhost|2703|informixtest_tcp|in
formix|0:CLDB:sysmaster

ONLN|2022-06-29
22:23:27.000|localhost|2714|informixtest_tcp|in
formix|0:OPDB:sysmaster:0:-

ONLN|2022-06-29
22:23:27.000|localhost|2714|informixtest_tcp|in
formix|0:CLDB:sysmaster

ONLN|2022-06-29
22:38:31.000|localhost|2791|informixtest_tcp|in
formix|0:OPDB:sysmaster:0:-
```

Starting Auditing

Either make changes to the audit configuration file and restart the instance

…and/or…

Use the **onaudit** command to modify the settings



Auditing Setup: Configuration File

Audit Configuration File: **$INFORMIXDIR/aaodir/adtcfg**

```
#ADTMODE is obsolescent - superseded by ADT_CLASSIC_ENABLED, ADT_DBSSO, ADT_DBSA
#ADTMODE        0                       # Auditing mode - 0, 1, 3, 5, 7
ADTPATH         $INFORMIXDIR/aaodir
                                        # Directory to contain audit trail files
ADTSIZE         50000                   # Maximum size of audit trail file (bytes)
ADTERR          0                       # Error handling modes.
ADTROWS         0                       # 0 - For legacy auditing
                                        # 1 - log audit tables
                                        # 2 - 1+log primary key

ADT_CLASSIC_ENABLED     false           # Classic mode auditing       (boolean)
ADT_DBSSO               false           # Mandatory auditing of DBSSO (boolean)
ADT_DBSA                false           # Mandatory auditing of DBSA  (boolean)

ADT_SYSLOG_ENABLED      false           # Audit to Syslog enabled?    (boolean)
ADT_SYSLOG_IDENTIFIER                   # Identifier - default DBSERVERNAME
ADT_SYSLOG_OPTIONS      LOG_NDELAY,LOG_NOWAIT
                                        # Comma-separated options: choose from:
                                        # LOG_ODELAY, LOG_NDELAY, LOG_PERROR,
                                        # LOG_CONS, LOG_NOWAIT, LOG_PID
ADT_SYSLOG_FACILITY     LOG_USER        # Facility - choose one from:
                                        # LOG_USER, LOG_LOCAL0 .. LOG_LOCAL7,
                                        # LOG_AUTH, LOG_AUTHPRIV, LOG_DAEMON
                                        # Not recommended but recognized:
                                        # LOG_SYSLOG, LOG_CRON, LOG_FTP,
                                        # LOG_KERN, LOG_LPR, LOG_MAIL,
                                        # LOG_NEWS, LOG_SYSLOG, LOG_UUCP
ADT_SYSLOG_PRIORITY     LOG_INFO        # Priority - choose one from:
                                        # LOG_EMERG, LOG_ALERT, LOG_CRIT,
                                        # LOG_ERR, LOG_WARNING, LOG_NOTICE,
                                        # LOG_INFO, LOG_DEBUG
```

Auditing Setup: Configuration File

< 14.10.xC6

ADTMODE

0 = auditing disabled

1 = auditing on; starts auditing for all sessions

3 = auditing on; audits DBSSO actions

5 = auditing on; audits database server administrator (DBSA) actions

7 = auditing on; audits DBSSO and database server administrator (DBSA) actions

Auditing Setup: Configuration File

>= 14.10.xC6

ADT_CLASSIC_ENABLED - auditing

0/OFF/FALSE/DISABLE/NO
1/ON/TRUE/ENABLE/YES

ADT_DBSSO – audit DBSSO actions

0/OFF/FALSE/DISABLE/NO
1/ON/TRUE/ENABLE/YES

ADT_DBSA – audit DBSA actions

0/OFF/FALSE/DISABLE/NO
1/ON/TRUE/ENABLE/YES

Auditing Setup: Configuration File

ADTPATH

Location of Audit Files

ADTSIZE

Max Size (in bytes) of Audit Files before generating a new one

Configure Auditing with onaudit

- Set location of audit files (ADTPATH):

  onaudit -p /logs/auditfiles

  (directory must exist)

Make sure that the filesystem used by ADTPATH has lots of space and is secure

- Set max size of audit files (ADTSIZE):

  onaudit -s 2097152 *[2 MB]*

Configure Auditing with onaudit

Enable auditing:

<14.10.xC6 (ADTMODE)

`onaudit -l 1` (3=DBSSO, 5=DBA, 7=DBSSO & DBA)

>=14.10.xC6

`onaudit -L on`	Enable regular auditing to file
`onaudit -E on`	Enable auditing to syslog
`onaudit -S on`	Enable auditing of DBSSO
`onaudit -A on`	Enable auditing of DBSA

Configure Auditing with onaudit

Review changes

```
onaudit -c


Onaudit -- Audit Subsystem Configuration Utility

Current audit system configuration:
    ADTMODE    = 0    (obsolescent)
    ADTERR     = 0
    ADTPATH    = /infaudit
    ADTSIZE    = 1048576
    Audit file = 4
    ADTROWS    = 0
    ENABLED    = 1
    DBSSO      = 1
    DBSA       = 1
ASL (Audit-to-Syslog) configuration:
    IDENTIFIER = myinformix
    OPTIONS    = LOG_CONS,LOG_NDELAY,LOG_NOWAIT,LOG_PERROR,LOG_PID
    FACILITY   = LOG_LOCAL7
    PRIORITY   = LOG_ALERT
    ENABLED    = 1
```

Auditing Virtual Processor

When Auditing is enabled, will see an **ADT** VP running

```
Virtual processor summary:
 class        vps        usercpu    syscpu     total
 cpu          1          3.10       1.06       4.16
 aio          5          0.26       0.72       0.98
 lio          1          0.03       0.20       0.23
 pio          1          0.04       0.16       0.20
 adm          1          0.29       0.66       0.95
 soc          1          0.39       0.93       1.32
 msc          1          0.00       0.01       0.01
 adt          1          0.05       0.16       0.21
 fifo         1          0.02       0.18       0.20
 total        13         4.18       4.08       8.26
```

Enabling Auditing

If “onaudit” was used, the configuration file name will have the *SERVERNUM* following it:

```
-rw-rw-r-- 1 informix informix 1120 Mar 16 09:33 adtcfg
-rw-rw-r-- 1 informix informix 1241 Mar 16 09:37 adtcfg.0
-rw-r--r-- 1 informix informix  908 Jul  2  2014 adtcfg.std
```

What to Audit – Audit Events

- ***Nothing*** is audited by default
- Need to specify what events to track and for which users (can be all users)
- Audit "events" are 4-character codes representing a database activity, e.g.

OPDB	Open Database
CRTB	Create Table
GRDB	Grant Database Access

https://www.ibm.com/docs/en/informix-servers/14.10?topic=security-audit-event-codes-fields

What to Audit – Audit Masks

- Audit “masks” specify which events to track for a user
- Built-in mask names are used to avoid having to create a mask for every user
  - _default
  - _require
  - _exclude
- The built-in masks are supplied empty – they do not include any audit events to begin with

What to Audit – Audit Masks

- How are the masks applied?
  - A user audit mask is applied first
  - If there is no user audit mask, then the audit events are obtained from the **_default** mask
  - The **_require** audit events are also tracked
  - The **_exclude** mask indicates events to NOT track, even if they are in the other masks *(including _default and _require)*

Make sure that a _default or _require mask is configured with a basic set of events so that they will be applied automatically for new users

What to Audit – Create Audit Masks

- Add a new Audit Mask: **onaudit -a**

- Create a basic audit mask for ***all*** users for opening a database *(OPDB)* and closing a database *(CLDB):*

```
onaudit -a -u _require -e +OPDB,CLDB
```

Audit Masks

Create an audit mask for user “jack” to track the creation *(CRTB)* and dropping *(DRTB)* of tables:

```
onaudit -a -u jack -e +CRTB,DRTB
```

Audit Masks - View

- Display the Audit Masks and their Audit Events:

```
onaudit -o -y

Onaudit -- Audit Subsystem Configuration Utility

_require          -        CLDB,OPDB
jack              -        CRTB,DRTB
```

- User “jack” has its own Audit Mask, but will inherit the **_require** events also

Audit Masks - Modify

Modify an existing Audit Mask: **onaudit -m**

Add Insert Row event (INRW) and remove the Drop Table event (DRTB) for audit mask “jack”:

```
onaudit -m -u jack -e +INRW -e -DRTB
```

Audit Masks – Add using Base Mask

Base an audit mask off of an existing mask, using "-r *<basemask>*"

Create a mask "jill" based off "jack", and add events Delete Row (DLRW) and Update Row (UPRW):

```
onaudit -a -u jill -r jack -e +DLRW,UPRW

onaudit -o -y
Onaudit -- Audit Subsystem Configuration Utility

_require                    -       CLDB,OPDB
jack                        -       CRTB,INRW
jill                        -       CRTB,DLRW,INRW,UPRW
```

Use Templates as Base Mask

Can create templates for different roles and use these as the base mask for new users

```
tmplt_rouser   -   OPDB,RDRW
tmplt_rwuser   -   OPDB,DLRW,INRW,UPRW

onaudit -a -u newbie -r tmplt_rwuser

newbie         -   DLRW,INRW,OPDB,UPRW
```

Audit Masks - Delete

Delete an existing Audit Mask: **onaudit -d**

Delete audit mask “jill”:

```
onaudit -d -u jill

onaudit -o -y
Onaudit -- Audit Subsystem Configuration Utility

_require                    -        GRDB,OPDB
jack                        -        CRTB,INRW
```

Audit Masks – Load from File

Instead of having to specify all of the individual events on the command line, put them in a file and load them. Makes changing the events simpler.

Create a text file in the format:

```
mask_name      base_mask      event_list
```

Recommend that the file be placed in $INFORMIXDIR/dbssodir

```
cd $INFORMIXDIR/dbssodir
cat event_list_all
_require -
ADCK,ADLG,ALFR,ALIX,ALLC,ALME,ALSQ,ALTB,ALTX,ALUR,CLDB,CRAG,CRAM,CRBS,CRBT,CRCT,CRDB,CRDS,CRD
T,CRIX,CRLB,CRLC,CRME,CROC,CRPL,CRPT,CRRL,CRRT,CRSN,CRSP,CRSQ,CRTB,CRTR,CRTX,CRUR,CRVW,CRXD,C
RXT,DLRW,DNCK,DNDM,DRAG,DRAM,DRBS,DRCK,DRCT,DRDB,DRDS,DRIX,DRLB,DRLC,DRLG,DRME,DROC,DRPL,DRRL
,DRRT,DRSN,DRSP,DRSQ,DRTB,DRTR,DRUR,DRTX,DRTY,DRVW,DRXD,DRXT,GRDB,GRDR,GRFR,GRLB,GRRL,GRSA,GR
SS,GRTB,GRXM,INRW,LGDB,LSAM,MDLG,ONAU,ONBR,ONCH,ONIN,ONLG,ONLO,ONMN,ONMO,ONPA,ONPL,ONSP,ONTP,
ONUL,OPDB,OPST,PWUR,RBSV,RLSV,RMCK,RNUR,RNDB,RNDS,RNIX,RNLB,RNLC,RNPL,RNSQ,RNTC,RNTX,RVDB,RVD
R,RVFR,RVLB,RVRL,RVSA,RVSS,RVTB,RVXM,STCO,STCN,STDF,STDP,STDS,STEP,STEV,STNC,STOM,STOP,STRL,S
TRS,STRT,STSA,STSC,STSN,STSV,STTX,SVXD,TCTB,UPAM,UPCK,UPDM,UPRW,USSP,USTB
```

(this is all on a single line)

onaudit -d -u _require ← Delete the mask if it already exists

onaudit -f event_list_all ← Load the new mask(s) from the file

Audit Masks & Events

- Take care when determining which events to audit
  - Too many may require an impractically large amount of storage, and files will become unmanageable, and performance may be impacted
  - Too few may leave gaps in the auditing and make it ineffective

- The audit events change between Informix versions – new ones added, obsolete ones removed

Audit File

- Audited events will be logged in a file created in the directory specified by ADTPATH

- The file will be named: *$INFORMIXSERVER.n*

```
-rw-rw---- 1 informix informix   356 Jun 29 22:38 informixtest_tcp.0
```

- Events can also be recorded in syslog (>=14.10.xC6)

Audit File – What's in it?

In the example audit file, we see the Open Database (OPDB) and Close Database (CLDB) events:

```
ONLN|2022-07-09
19:52:18.000|localhost|4464|informixtest_tcp|jack
|0:OPDB:stores_demo:0:-
ONLN|2022-07-09
19:52:18.000|localhost|4464|informixtest_tcp|jack
|0:CLDB:stores_demo
```

No information on what "jack" did while connected to the database, as these were not specified in an audit mask

Make sure all events you want to audit are included in the mask

Audit File – What's in it?

```
ONLN|2022-07-09 19:52:18.000|
localhost|4464|informixtest_tcp|jack
|0:OPDB:stores_demo:0:-
```

- The file contains a pipe delimited set of fields:
  - ONLN
  - DateTime
  - Hostname
  - PID
  - DB Server Name
  - User Name

Generic application IDs, for example, connections from App Server/Web Server, will not show you *who* did what

- The last field shows information on the event, delimited by colon:
  - Error Code
  - Event Code (4-character audit event)
  - Variable fields, depends on the event code

Audit Record Example

```
ONLN|2022-07-09 19:52:18.000|
localhost|4464|informixtest_tcp|jack|
0:OPDB:stores_demo:0:-
```

- The event error code (0=Success)
- The event code (OPDB=Open Database)
- The **OPDB** *(Open Database)* event entry shows:
  - Database Name
  - Exclusive Flag
  - Database Password

Audit Record Example

```
ONLN|2022-07-09 19:56:17.000|
localhost|4477|informixtest tcp|jack|
0:CRTB:stores_demo:160:mytab:jack:0:-
```

- The **CRTB** *(Create Table)* event entry shows:
  - Database Name
  - Tab ID
  - Table Name
  - Table Owner
  - Fragmentation Flags *[0=Not Fragmented, 1=In DBSpace, etc]*
  - DBSpace List

The fields displayed for the event vary by the audit event – complicates reporting

Audit File - Example

```
ONLN|2022-07-09 20:22:24.000|localhost|4525|informixtest_tcp|jack|-329:OPDB:nodb:0:-

ONLN|2022-07-09 20:22:24.000|localhost|4525|informixtest_tcp|jack|0:OPDB:stores_demo:0:-

ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:CRTB:stores_demo:162:mytab:jack:1:rootdbs

ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:INRW:stores_demo:162:1049
ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:INRW:stores_demo:162:1049
ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:INRW:stores_demo:162:1049096:259::

ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:UPRW:stores_demo:162:1049096:258:1049096:258::

ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:DLRW:stores_demo:162:1049096:257::
ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:DLRW:stores_demo:162:1049096:258::
ONLN|2022-07-09
20:22:24.000|localhost|4525|informixtest_tcp|jack|0:DLRW:stores_demo:162:1049096:259::
ONLN|2022-07-09

20:22:24.000|localhost|4525|informixtest_tcp|jack|0:DRTB:stores_demo:162:mytab:jack:0:1049096
ONLN|2022-07-09 20:22:24.000|localhost|4525|informixtest_tcp|jack|0:CLDB:stores_demo
```

Error Code

DBSpace Name

Row IDs

Multiple Deletes

Does NOT track anything about the *data* that was Inserted, *what* was Updated, *what* was Deleted or the *SQL* that was executed

Audit to syslog (ASL)

If **ADT_SYSLOG_ENABLED** is true, auditing events are placed in the syslog file.
Format is the same as in the auditing file:

>= 14.10.xC6

/var/log/syslog:

```
Jul 10 19:51:20 informixtest myinformix[1188]: ONLN|2022-07-10
19:51:19.000|localhost|1562|informixtest_tcp|informix|0:STSN
Jul 10 19:51:20 informixtest myinformix[1188]: ONLN|2022-07-10
19:51:19.000|localhost|1562|informixtest_tcp|informix|0:OPDB:sysma
ster:0:-
Jul 10 19:51:20 informixtest myinformix[1188]: ONLN|2022-07-10
19:51:19.000|localhost|1562|informixtest_tcp|informix|0:ONAU:-c
Jul 10 19:51:20 informixtest myinformix[1188]: ONLN|2022-07-10
19:51:19.000|localhost|1562|informixtest_tcp|informix|0:INRW:sysma
ster:232:1026:7436::
Jul 10 19:51:20 informixtest myinformix[1188]: ONLN|2022-07-10
19:51:19.000|localhost|1562|informixtest_tcp|informix|0:CLDB:sysma
ster
```

In this example, **ADT_SYSLOG_IDENTIFIER** set to “myinformix”

Audit to syslog (ASL)

Options in adtcfg configuration file:

>= 14.10.xC6

```
ADT_SYSLOG_ENABLED      false          # Audit to Syslog enabled?    (boolean)
ADT_SYSLOG_IDENTIFIER                  # Identifier - default DBSERVERNAME

ADT_SYSLOG_OPTIONS      LOG_NDELAY,LOG_NOWAIT
                                       # Comma-separated options: choose from:
                                       # LOG_ODELAY, LOG_NDELAY, LOG_PERROR,
                                       # LOG_CONS, LOG_NOWAIT, LOG_PID
ADT_SYSLOG_FACILITY     LOG_USER       # Facility - choose one from:
                                       # LOG_USER, LOG_LOCAL0 .. LOG_LOCAL7,
                                       # LOG_AUTH, LOG_AUTHPRIV, LOG_DAEMON
                                       # Not recommended but recognized:
                                       # LOG_SYSLOG, LOG_CRON, LOG_FTP,
                                       # LOG_KERN, LOG_LPR, LOG_MAIL,
                                       # LOG_NEWS, LOG_SYSLOG, LOG_UUCP
ADT_SYSLOG_PRIORITY     LOG_INFO       # Priority - choose one from:
                                       # LOG_EMERG, LOG_ALERT, LOG_CRIT,
                                       # LOG_ERR, LOG_WARNING, LOG_NOTICE,
                                       # LOG_INFO, LOG_DEBUG
```

Check man syslog for information on these options

Audit to syslog (ASL)

ADT_SYSLOG_OPTIONS, ADT_SYSLOG_FACILITY, ADT_SYSLOG_PRIORITY are not shown in the syslog

Can be used to identify auditing entries

Can use **journalctl** to filter logged auditing entries

ADT_SYSLOG_FACILITY	LOG_LOCAL7
ADT_SYSLOG_PRIORITY	LOG_ALERT

```
journalctl --priority alert --facility local7

-- Logs begin at Wed 2022-06-29 22:58:28 UTC, end at Sun 2022-07-10 20:30:19
UTC. --
Jun 30 13:50:44 informixtest myinformix[1269]: ONLN|2022-06-30
13:50:44.000|localhost|1562|informixtest_tcp|informix|0:CLDB:sys>
Jun 30 13:50:47 informixtest myinformix[1269]: ONLN|2022-06-30
13:50:47.000|localhost|1563|informixtest_tcp|informix|0:OPDB:sys>
Jun 30 13:50:47 informixtest myinformix[1269]: ONLN|2022-06-30
13:50:47.000|localhost|1563|informixtest_tcp|informix|0:CLDB:sys>
Jun 30 13:51:09 informixtest myinformix[1269]: ONLN|2022-06-30
13:51:09.000|localhost|1564|informixtest_tcp|informix|0:OPDB:sys>
```

Row Level Auditing

- Tracking ***all*** Inserts, Updates, Deletes and even Selects against ***all*** tables may not be practical
- Set **Row Level Auditing** level to restrict auditing of the following Events to only those tables that have been set to "audit"
  - DLRW – Delete Row
  - INRW – Insert Row
  - RDRW – Read Row
  - UPRW – Update Row
- All other events still apply to all tables

Row Level Auditing

Set Row Level Auditing (ADTROWS):

onaudit -R [0|1|2]

- 0 Audit Row Level events on all tables
- 1 Only track DLRW, INRW, RDRW, UPRW for tables with auditing set
- 2 Same as 1, but record any integer primary key in the audit file

Row Level Auditing

```
create table tab1(a int) with audit;
```

Row level events will be audited for the *new* table

```
alter table tab2 add audit;
```

Row level events will be audited for an *existing* table

```
alter table tab drop audit;
```

Remove auditing from a table

Audit Files

- The Audit Files are in the directory specified by ADTPATH in the auditing config
- The files are named $INFORMIXSERVER.*n*
- When the file size (in bytes) reaches the value set by ADTSIZE, audit records are written to a new file with the next number
- When the instance is restarted, a new file is created
- Force a new file with **onaudit -n**

The “current” Audit File

Current Audit File Number in file
$INFORMIXDIR/aaodir/adtlog.*<SERVERNUM>*

```
-rw-rw---- 1 informix informix    2 Mar 16 18:59 adtlog.0
```

```
cat adtlog.0
2
```

The current file is also shown in onaudit -c

Audit Files

- The audit files will need to be purged/archived periodically
- Need a strategy for dealing with the audit files, for example:
  - Keep 6 months of files
  - Move older files to another filesystem and compress them
  - Remove compressed files after 12 months

Establish a retention period for the audit files, or allocate lots of space!

onshowaudit

- Use onshowaudit to view the audit files
- By default, shows the contents of all available audit files, not just the latest
- If used *without* the -n or -f option, uses the ADTPATH from the adtcfg file, and not from the adtcfg.<*SERVERNUM*> configuration file

```
The directory name specified by the ADTPATH
configuration parameter does not
exist or does not have the necessary permissions.
```

onshowaudit

- Use the **-f *<filename>*** parameter to show the contents of an individual, named audit file
- Use the **-n *<SERVERNUM>*** parameter to show the contents of the audit files for the supplied server
- Use the **-u *<user>*** and **-s *<servername>*** to limit results to the user or server supplied

onshowaudit

- Use **-l [<*filename*>]** to format the output with "pipe" delimiters for the audit event specific fields
- The optional filename puts the results in the named file, so it can then be loaded into a table or parsed more easily

Loading the audit information into a table makes it easier to store and report on…but is that really what you want to do?
Consider security issues and data volume!

onshowaudit

onshowaudit -n 0 -d -u jack -f informixtest_tcp.6

```
ONSHOWAUDIT Secure Audit Utility
INFORMIX-SQL Version 14.10.FC7
ONLN|2022-07-09 19:52:18.000|localhost|4464|informixtest_tcp|jack|0:OPDB:stores_demo:0:-
ONLN|2022-07-09 19:52:18.000|localhost|4464|informixtest_tcp|jack|0:CLDB:stores_demo
ONLN|2022-07-09 19:55:58.000|localhost|4477|informixtest_tcp|jack|0:OPDB:stores_demo:0:-
ONLN|2022-07-09
19:56:17.000|localhost|4477|informixtest_tcp|jack|0:CRTB:stores_demo:160:mytab:jack:0:-
ONLN|2022-07-09
19:56:17.000|localhost|4477|informixtest_tcp|jack|0:DRTB:stores_demo:160:mytab:jack:0:1049096
```

onshowaudit -n 0 -u jack -f informixtest_tcp.6 -l -q

```
ONLN|2022-07-09 19:52:18.000|localhost|4464|informixtest_tcp|jack|0|OPDB|stores_demo|||||||0|-|
ONLN|2022-07-09 19:52:18.000|localhost|4464|informixtest_tcp|jack|0|CLDB|stores_demo||||||||
ONLN|2022-07-09 19:55:58.000|localhost|4477|informixtest_tcp|jack|0|OPDB|stores_demo|||||||0|-|
ONLN|2022-07-09
19:56:17.000|localhost|4477|informixtest_tcp|jack|0|CRTB|stores_demo|160|mytab||||jack|0|-|
ONLN|2022-07-09
19:56:17.000|localhost|4477|informixtest_tcp|jack|0|DRTB|stores_demo|160|mytab||||jack|0|104909
6|
```

No blank lines in the output with "-l", or if "-d" is specified (>=14.10.xC6)

Demonstration of Informix Auditing

Role Separation

- ***Without Role Separation***, the **informix** user can stop auditing, change audit masks, mess with the audit files…
  - Undermines the effectiveness of auditing

- ***With Role Separation***, only specific users can change the auditing configuration, change events that are audited and perform the Informix administration

- Allows you to cut down use of the informix account and reserve it for special occasions only

Role Separation

- **Audit Analysis Officer (AAO)**
  - Configure auditing
  - Review auditing information
  - Manage audit files
- **Database System Security Officer (DBSSO)**
  - Modify Audit Masks
- **Database Server Administrator (DBSA)**
  - Perform database maintenance

Enable Role Separation

- Role Separation requires discrete UNIX ***groups*** to be set up for each role: AAO, DBSO, DBSA
- Add one or more user accounts to each group
- Can use your own names for groups/IDs
- Avoid overlapping of roles, but it is allowed
- Set up Role Separation at Installation Time or after install

Enable Role Separation

- At install time
  - Make sure choose Custom install, not Typical
- Will be prompted to enter the:
  - Group for security related tasks (DBSSO role)
  - Group for audit administration tasks (DBSSO role)
  - Group for database users.

Enable Role Separation

- To enable role separation *after* install, then change the group of the aaodir, dbssodir and etc directories under INFORMIXDIR to the role groups for AAO, DBSSO and DBSA, e.g.

```
drwxrwxr-x 2 informix ifxaao   4096 Mar 23 11:30 aaodir
drwxrwxr-x 2 informix ifxdbsso 4096 Mar 22 22:38 dbssodir
drwxrwxr-x 4 informix ifxdbsa  4096 Mar 23 11:43 etc
```

- Change permissions for oninit

```
chmod 6755 oninit
```

- Change group of ONCONFIG and sqlhosts to the DBSA group

Role Separation - Examples

```
informix@pigriffin:~ $ onaudit -l 0
Onaudit -- Audit Subsystem Configuration Utility

Must be an AAO or DBSSO to run this program.
```

```
ifxdbsso@pigriffin:~ $ onshowaudit -n 1

Must be a DBSA, user root or an AAO to run this program
```

```
ifxaao@pigriffin~ $ onaudit -o -y
Onaudit -- Audit Subsystem Configuration Utility

Must be a DBSSO to execute this action.
```

```
ifxdbsa@pigriffin:~ $ onshowaudit -n 1

Must be an AAO to run this program.
```

Informix Auditing – Questions to Ask

- Where to store the audit files?
  - Space?
- What to do with the audit files?
  - Retention? Archiving? Store in the database?
- What events to capture?
- Is it practical to audit all inserts, updates, deletes, etc?
  - Row Level Auditing?
- Does it record enough detail?
  - Schema and data change details are limited
  - SQL is not recorded

Informix Auditing – Questions to Ask

- Is role separation necessary?
  - Who are the AAO, DBSO, DBSA users?
- If using Row Level Auditing, how to verify that it is still in place?
- What to do with the recorded events?
  - Format makes it hard to report on.
  - Use for review "after the fact"?
  - Monitor for specific events and trigger an alert?

Does Informix auditing solve "the problem"?

Other Auditing Methods

- Guardium
- Capture running SQL:
  - SQL Trace
  - Third Party Tools to track SQL submitted
- Table changes:
  - Triggers
  - Dump out schemas and compare

Advanced Informix Consulting and Support

- **Informix Remote DBA 24/7** Peace of mind for your systems
- **Expert consultants** for any Informix problem
- Support for **Informix Upgrades** from any version
- **Migrations** to new hardware, let us help virtualize your systems
- Get help **configuring** and **managing** UNIX systems
- Informix **cloud** migrations
- **IBM Informix sales**
- Let us **tune your system**, we can maximize the potential of your database
- *What can we do for you today?*

xDB SYSTEMS

Questions?

Send follow-up questions to
mike@xdbsystems.com

Thank You

Mike Walker

mike@xdbsystems.com

For more information:

https://www.xdbsystems.com