Best Practices: Informix Auditing

Mike Walker

Advanced DataTools Corporation

mike@advancedatatools.com

Mike Walker

Mike Walker has been using Informix databases for over 20 years, as a developer and as a database administrator.

Mike heads up the Remote DBA Support for Advanced DataTools Corporation.

Contact Info:
mike@advancedatatools.com
www.advancedatatools.com
Office: 703-256-0267
Cell: 303-909-4265

Informix Auditing – What is it?

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

Auditing Setup: Configuration File

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

```
#*************************************************************************
#
#   Licensed Material - Property Of IBM
#
#   "Restricted Materials of IBM"
#
#   IBM Informix Dynamic Server
#   (c) Copyright IBM Corporation 1996, 2008 All rights reserved.
#
#   Title:       adtcfg
#   Description: IBM INFORMIX Dynamic Server Audit Configuration file.
#                IBM IDS will read this file when a server is either
#                initialized or restarted and will configure the audit
#                subsystem according to the values herein. Audit
#                Analysis Officer has the responsibility of updating
#                this file with values suitable for the specific instance.
#
#                IBM INFORMIX Dynamic Server will write the file
#                adtcfg.<server_number> with any changes to the values
#                of these parameters within the instance.
#
#*************************************************************************

ADTMODE         0                       # Auditing mode
ADTPATH         /usr/informix/aaodir    # Directory where audit trails will be written by OnLine
ADTSIZE         50000                   # Maximum size of any single audit trail file
ADTERR          0                       # Error handling modes.
```

Auditing Setup: Configuration File

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

…and/or…

Use the onaudit command to modify the settings

Use of onaudit command copies the settings and then uses a configuration file named adtcfg.<*servernum*>

Auditing Setup: Configuration File

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 actions

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

Do you want *extra* auditing for the privileged accounts?

Auditing Setup: Configuration File

ADTPATH

Location of Audit Files

ADTSIZE

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

Auditing Setup: Configuration File

ADTERR

- *0 = continue error mode*

  When it encounters an error as it writes an audit record, the database server writes a message of the failure into the message log. It continues to process the thread.

- *1 = halt error mode: suspend thread processing*

  When the database server encounters an error as it writes an audit record, the database server suspends processing of the thread until it successfully writes a record.

- *3 = halt error mode: shut down system*

  When the database server encounters an error as it writes an audit record, the database server shuts down.

Avoid ADTERR=0 if you want to make sure that no auditable events are missed

Auditing Setup: Configuration File

ADTROWS – Row Level Auditing

(Not included in the supplied adtcfg file)

- 0 for auditing row-level events on all tables
- 1 to allow control of which tables are audited. Row-level events (Delete-Row, Insert-Row, Read-Row, Update-Row) are audited only on tables for which the AUDIT flag is set.
- 2 to turn on selective row-level auditing and also to include the primary key in audit records (the primary key is only recorded if it is an integer)

onaudit

```
Usage: onaudit <action> [-f file] [-u name] [-r bmsk] [-e eset] [-y]
       onaudit [-c] [-n] [-l lev] [-e err] [-p path] [-s size]

action: one of
    -a    -- add a mask
    -d    -- delete a mask
    -m    -- modify a mask
    -o    -- output a mask
 -r bmsk -- name of basemask
 -c      -- print audit configuration
 -n      -- start new log file
 -l lev  -- set ADTMODE
 -e err  -- set ADTERR
 -p path -- set ADTPATH
 -s size -- set ADTSIZE
 -f file -- include instruction file
 -u mask -- name of target/mask
 -e eset -- event set added to (+) or removed from (-) mask
 -R fga  -- set ADTROWS for Fine-Grained Auditing
 -y      -- respond yes to all prompts
```

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

Configure Auditing with onaudit

- Set max size of audit files (ADTSIZE):

  `onaudit -s 2097152` *[2 MB]*

- Enable auditing (ADTMODE):

  `onaudit -l 1`

Configure Auditing with onaudit

Review changes

```
onaudit -c

Onaudit -- Audit Subsystem Configuration Utility

Current audit system configuration:
    ADTMODE    = 1
    ADTERR     = 0
    ADTPATH    = /logs/auditfiles
    ADTSIZE    = 2097152
    Audit file = 0
    ADTROWS    = 0
```

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

Informix Log File:

```
09:37:02  Dynamically added 1 adt VP
09:37:02  Audit Mode changed to 1
```

Because on “onaudit” was used, the configuration file name now has 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/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sec.doc/ids_au_104.htm#ids_au_104

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 the database *(OPDB)* and granting database permissions *(GRDB):*

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

Audit File

- As user “jack”, create and drop a table:

```
create table mytab(a serial);
drop table mytab;
```

- 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 162 Mar 16 09:51 /logs/auditfiles/griffin.0
```

Audit File – What's in it?

- Look at the audit file, we ***only*** see the Open Database (OPDB) event:

```
ONLN|2017-03-16 09:51:56.000|
pigriffin|6232|griffin|jack|
0:OPDB:stores_demo:0:-
```

No information on what "jack" did – those *events* (create table, drop table) 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|2017-03-16 09:51:56.000|
pigriffin|6232|griffin|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|2017-03-16 09:51:56.000|
pigriffin|6232|griffin|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 Masks

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

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

Audit Masks - View

- Display the Audit Masks and their Audit Events:

```
onaudit -o -y

Onaudit -- Audit Subsystem Configuration Utility

_require                    -       GRDB,OPDB
jill                        -       CRTB,DRTB
```

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

Audit Masks

As user “jill”, create and drop a table:

```
create table mytab(a serial);
insert into mytab(a) values (0);
drop table mytab;
```

Audit File

Now the audit file contains additional events:

```
ONLN|2017-03-16 10:59:32.000|
pigriffin|6326|griffin|jill|0:OPDB:stores_demo:0:-

ONLN|2017-03-16 10:59:32.000|
pigriffin|6326|griffin|jill|0:CRTB:stores_demo:115:myt
ab:jill:0:-

ONLN|2017-03-16 10:59:32.000|
pigriffin|6326|griffin|jill|0:DRTB:stores_demo:115:myt
ab:jill:0:5242978
```

Audit Record Example

```
ONLN|2017-03-16 10:59:32.000|
pigriffin|6326|griffin|jill|
0:CRTB:stores_demo:115:mytab:jill: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 Masks - Modify

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

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

```
onaudit -m -u jill -e +INRW -e -DRTB
Onaudit -- Audit Subsystem Configuration Utility

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

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

Audit Masks – Add using Base Mask

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

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

```
onaudit -a -u jack -r jill -e +DLRW,UPRW
Onaudit -- Audit Subsystem Configuration Utility

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

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

Audit Masks – 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 -- Audit Subsystem Configuration Utility

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

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

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 – Load from File

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

_require                                        -
ADCK,ADLG,ALIX,ALTB,STSV,CLDB,CRAM,CRBS,CRDB,CRDS,CRIX,CRSN
,CRSP,CRTB,CRTR,CRVW,DLRW,DNCK,DNDM,DRAM,DRBS,DRCK,DRDB,DRD
S,DRIX,RLSV,DRLG,DRSN,DRSP,DRTB,DRTR,DRVW,GRDB,GRTB,INRW,LG
DB,LSAM,MDLG,RVLB,ONAU,ONCH,ONIN,ONLG,ONLO,ONMO,ONPA,GRSS,O
NSP,ONTP,ONUL,OPDB,GRSA,RVSA,RMCK,RNTC,RVDB,RVTB,RVSS,STCN,
STDF,STSN,UPAM,UPCK,UPDM,UPRW,USSP,USTB,ALFR,STDS,STTX,STOM
,STRT,STOP,GRFR,RVFR,CRRL,DRRL,GRRL,RVRL,STDP,STRL,STSA,ONM
N,RNDB,ONBR,ONPL,OPST,CRRT,DRRT,CRDT,CRCT,DRCT,CRBT,DRTY,CR
ME,DRME,ALME,CROC,DROC,RBSV,STRS,CRAG,DRAG,STSC,RVXM,RNIX,C
RSQ,RNSQ,DRSQ,ALSQ,STEV,RNDS,GRDR,RVDR,STCO,STNC,STEP,CRPT,
CRXT,CRXD,DRXT,DRXD,TCTB,SVXD,CRLC,CRPL,CRLB,DRLC,DRPL,DRLB
,RNLC,RNPL,RNLB,ALLC,GRXM,GRLB,CRTX,ALTX,RNTX,DRTX,CRUR,ALU
R,DRUR,RNUR,PWUR
jack                                            -
CRTB,DLRW,INRW,UPRW
```

Audit File - Example

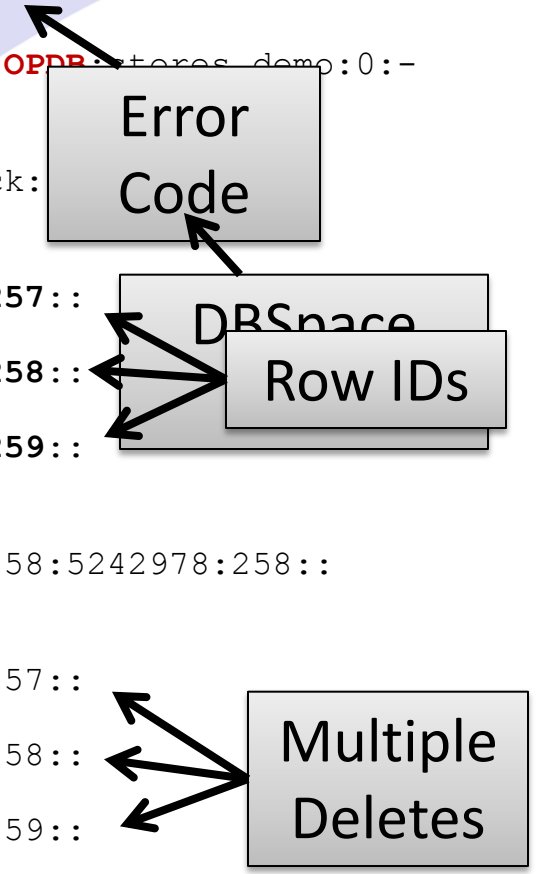
```
database nodb;
```
← This database does not exist

```
database stores_demo;

create table mytab(a serial, b char(1)) in datadbs;

insert into mytab(a,b) values (0,"A");
insert into mytab(a,b) values (0,"B");
insert into mytab(a,b) values (0,"C");

update mytab set b="X" where b="B";

delete from mytab where 1=1;
```
← 1 statement, deletes 3 records

```
drop table mytab;
```

Audit File - Example

```
ONLN|2017-03-16 12:47:23.000|pigriffin|6454|griffin|jack|-329:OPDB:nodb:0:-

ONLN|2017-03-16 12:51:00.000| pigriffin|6454|griffin|jack|0:OPDB:stores_demo:0:-

ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:CRTB:stores_demo:116:mytab:jack:

ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:INRW:stores_demo:116:5242978:257::
ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:INRW:stores_demo:116:5242978:258::
ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:INRW:stores_demo:116:5242978:259::

ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:UPRW:stores_demo:116:5242978:258:5242978:258::

ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:DLRW:stores_demo:116:5242978:257::
ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:DLRW:stores_demo:116:5242978:258::
ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:DLRW:stores_demo:116:5242978:259::

ONLN|2017-03-16 12:51:00.000|
pigriffin|6454|griffin|jack|0:DRTB:stores_demo:116:mytab:jack:0:5242978
```



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

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

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: **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
- Modifies ADTROWS in the configuration
- Set Row Level Auditing to only track tables with auditing enabled:

```
onaudit -R 2
Onaudit -- Audit Subsystem Configuration Utility
```

Row Level Auditing

```
database stores_demo;

create table tab1(a int);
```

Row level events will *not* be audited for tab1

```
create table tab2(a int) with audit;
```

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

```
create table tab3(a int);
alter table tab3 add audit;
```

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

```
alter table tab3 drop audit;
```

Remove auditing from a table

Row Level Auditing - Example

```
grant insert on tab1 to public;
Permission granted.
```

tab1 has *NOT* been set to audit row level events

```
insert into tab1(a) values (1);
1 row(s) inserted.
```

```
insert into tab2(a) values (1);
1 row(s) inserted.
```

```
update tab2 set a=2 where a=1;
1 row(s) updated.
```

tab2 has been set to audit row level events

```
delete from tab2 where a=2;
1 row(s) deleted.
```

Row Level Auditing - Example

```
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:STSN
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:OPDB:stores_demo:0:-
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:GRTB:stores_demo:120:2:jack
:public:
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:INRW:stores_demo:121:524297
9:257::
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:UPRW:stores_demo:121:524297
9:257:5242979:257::
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:DLRW:stores_demo:121:524297
9:257::
ONLN|2017-03-16 15:55:29.000|
pigriffin|6604|griffin|jack|0:CLDB:s
```

Grant recorded for tab1, but Insert was not

Insert, Update and Delete WERE recorded for tab2

Row Level Auditing

- Enabling Row Level Auditing may create gaps in your auditing
- Need to review new tables for auditing needs, and changing business requirements may change the need for auditing

Include auditing requirement assessment as part of the development/design process

The "with audit" keywords will only be included in the table definition (dbexport/dbschema) when using the "-ss" option

Row Level Auditing - Alters

Be aware that *altering* a table that has been set to audit row level events removes the auditing!

```
insert into tab2(a) values (10);
alter table tab2 add b char(1);
insert into tab2(a) values (11);
delete from tab2 where 1=1;
```

Row Level Events following the ALTER are not recorded

```
ONLN|2017-03-16 16:51:10.000|
pigriffin|6644|griffin|jack|0:INRW:stores_demo:121:5242979:257::
ONLN|2017-03-16 16:51:10.000|
pigriffin|6644|griffin|jack|0:ALTB:stores_demo:121:121:5242979
```

Row Level Auditing - Alters

When run an ALTER, use “add audit” to retain auditing

```
insert into tab2(a) values (10);
alter table tab2 add b char(1), add audit;
insert into tab2(a) values (11);
delete from tab2 where 1=1;
```

```
ONLN|2017-03-16 17:21:19.000|
pigriffin|6653|griffin|jack|0:INRW:stores_demo:123:5242979:257::
ONLN|2017-03-16 17:21:19.000|
pigriffin|6653|griffin|jack|0:ALTB:stores_demo:123:123:5242979
ONLN|2017-03-16 17:21:19.000|
pigriffin|6653|griffin|jack|0:INRW:stores_demo:123:5242979:513::
ONLN|2017-03-16 17:21:19.000|
pigriffin|6653|griffin|jack|0:DLRW:stores_demo:123:5242979:257::
ONLN|2017-03-16 17:21:19.000|
pigriffin|6653|griffin|jack|0:DLRW:stores_demo:123:5242979:513::
```

Need to make sure that your auditable tables are still being audited

Row Level Auditing

Show tables that have auditing enabled

```
select tabname[1,30], flags
from systables
where flags != 0
  and decode(bitand(flags, 64),0,0,1) = 1
order by 1;

tabname                                flags

tab2                                      64
tab3                                      64
tab4                                      80
```

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**

Audit Files

Switch Audit File: **onaudit -n**

Message written to Informix Log File:

```
18:59:28  Audit trail switched to /logs/auditfiles/griffin.2
```

Multiple Audit Files:

```
-rw-rw---- 1 informix informix 13336 Mar 16 18:57 griffin.0
-rw-rw---- 1 informix informix   223 Mar 16 18:59 griffin.1
-rw-rw---- 1 informix informix   149 Mar 16 18:59 griffin.2
```

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!

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” Audit File

Shown in the Configuration: onaudit -c

```
Current audit system configuration:
    ADTMODE     = 1
    ADTERR      = 0
    ADTPATH     = /logs/auditfiles
    ADTSIZE     = 2097152
    Audit file = 2
    ADTROWS     = 2
```

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

```
USAGE:
   onshowaudit [-I | -O] [-f <input file>] [-u <user name>]
               [-s <server name>]
   onshowaudit [-n <server number>] [-l [<loadfile>]]

         -I : read from the Informix audit trail
         -O : read from the OS audit trail
         -f : read a single audit trail file (Informix mode only)
         -u : extract only records concerning <user name>
         -s : extract only records concerning <server name>
         -n : extract audit records from ADTPATH in
              ADTCFG.<servernumber>
         -l [<loadfile>] : print audit records with delimiters
```

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 -f griffin.1

ONLN|2017-03-16
18:59:28.000|pigriffin|6751|griffin|informix|0:STSN

ONLN|2017-03-16
18:59:28.000|pigriffin|6751|griffin|informix|0:OPDB:sys
master:0:-

onshowaudit -n 0 -f griffin.1 -l

ONLN|2017-03-16
18:59:28.000|pigriffin|6751|griffin|informix|0|STSN||||
||||||
ONLN|2017-03-16
18:59:28.000|pigriffin|6751|griffin|informix|0|OPDB|sys
master|||||||0|-|
```

No blank lines in the output with “-l”

What to do with the Audit Info?

- Why do you want to capture the auditing events?
  - Track what users did?
  - Look for potential security violations?
  - Find out who “changed” something?
- Do you want to be ***proactive*** or ***reactive***?
- Consider monitoring/parsing the audit files and generating alerts for particular events, e.g. User *x* ALTERED table *y* @ *hh:mm*

Audit Alerts

Send email when identify events of interest

Create Table Events [CRTB]								
Date	Hostname	PID	Username	ErrNo	Database	Tab ID	Table Name	Owner
04/01/2017 18:03:07	pigriffin	32023	informix	0	tstres	100	miketest	informix
04/04/2017 13:28:06	pigriffin	3540	informix	0	stores_demo	155	miketest	informix
04/04/2017 13:29:53	pigriffin	3540	informix	0	stores_demo	156	miketest	informix
04/04/2017 13:30:19	pigriffin	3540	informix	0	stores_demo	157	miketest	informix
04/04/2017 13:30:28	pigriffin	3540	informix	0	stores_demo	158	miketest	informix
04/04/2017 13:30:55	pigriffin	3540	informix	0	stores_demo	159	miketest	informix
04/04/2017 13:32:21	pigriffin	3540	informix	0	stores_demo	160	miketest	informix
04/04/2017 13:32:40	pigriffin	3540	informix	0	stores_demo	161	miketest	informix

Drop Table Events [DRTB]								
Date	Hostname	PID	Username	ErrNo	Database	Tab ID	Table Name	Owner
04/04/2017 13:29:53	pigriffin	3540	informix	0	stores_demo	155	miketest	informix
04/04/2017 13:30:19	pigriffin	3540	informix	0	stores_demo	156	miketest	informix
04/04/2017 13:30:28	pigriffin	3540	informix	0	stores_demo	157	miketest	informix
04/04/2017 13:30:55	pigriffin	3540	informix	0	stores_demo	158	miketest	informix
04/04/2017 13:32:21	pigriffin	3540	informix	0	stores_demo	159	miketest	informix
04/04/2017 13:32:40	pigriffin	3540	informix	0	stores_demo	160	miketest	informix

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

Role Separation

	informix	DBSA	AAO	DBSSO
Start/Stop Auditing (onaudit -l)	×	×	✓	×
View Audit Configuration (onaudit -c)	×	×	✓	×
View Auditing Logs (onshowaudit)	×	×	✓	×
View/Modify Audit Masks (onaudit -a, -d, -m, -o)	×	×	×	✓
View/Modify Audit Files on Disk	✓*	×	✓	×
Start/Stop Instance	✓ **	✓	×	×
Add/Drop DBSpace	✓	✓	×	×
Run dbaccess/onstat	✓	✓	✓	✓
Connect to database with no explicit permissions	✓	×	×	×
Remove chunk cooked files	✓	×	×	×
onbar	✓	×	×	×

* Can limit with UNIX permissions on audit file directory

** Not recommended

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)

```
Get Role Separation choice
--------------------------

Enable role separation for auditing procedures.

If you enable role separation, you can assign existing groups of
users to
specific roles.

If you do not enable role separation, the database server
administrator
performs all administration tasks.

    1- Enable role separation
  ->2- Do not enable role separation

ENTER THE NUMBER FOR YOUR CHOICE, OR PRESS <ENTER> TO ACCEPT THE
DEFAULT:: 1
```

Enable Role Separation

Prompted to enter the groups names for DBSSO and AAO

```
Role Separation groups selection
--------------------------------

Assign a group of users to each of the following roles by specifying
group
identifiers (group IDs). The group IDs specified must already exist on
your
system.

Group for security-related tasks: (Default: informix): ifxdbsso

Group for audit-administration tasks: (Default: informix): ifxaao

Group for database users (leave blank to allow all users): (Default: ):
```

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”?

Webcasts
Informix Best Practices Series

- Getting Started with Informix
- Informix Configuration - Part 1
- Informix Configuration - Part 2
- Disks & Database Layout
- Backup, Recovery, and High Availability Disaster Recovery
- Informix Connection Manager
- Informix Auditing

Replays available at:
http://advancedatatools.com/Informix/Webcasts.html

Next Webcast

Running Informix in a Monster Virtual Machine

August 31st, 2 PM EDT

Lester Knutsen

Register:
http://advancedatatools.com/Informix/NextWebcast.html

Informix Resources - IIUG

- The International Informix User Group
  - www.iiug.org
  - Membership is FREE
- Washington Area Informix User Group
  - waiug.org
  - Next meeting: August 8th, 8:30-1:00, McLean, VA

Informix Training in 2017

- September 11-14, 2017
  - **Advanced Informix Performance Tuning**
- September 18-21, 2017
  - **Informix for Database Administrators**
- All courses can be taken online on the web from your desk or at our training center in Virginia.
- We guarantee to *NEVER* cancel a course and will teach a course as long as one student is registered!

Questions?

Send follow-up questions to
mike@advancedatatools.com

Informix Support and Training from the Informix Champions!

Advanced DataTools is an Advanced Level IBM Informix Data Management Partner, and has been an authorized Informix partner since 1993. We have a long-term relationship with IBM, we have priority access to high-level support staff, technical information, and Beta programs. Our team has been working with Informix since its inception, and includes 8 Senior Informix Database Consultants, 4 IBM Champions, 2 IIUG Director's Award winners, and an IBM Gold Consultant. We have Informix specialists Lester Knutsen and Art Kagel available to support your Informix performance tuning and monitoring requirements!

- ***Informix Remote DBA Support Monitoring***
- ***Informix Performance Tuning***
- ***Informix Training***
- ***Informix Consulting***
- ***Informix Development***

Free Informix Performance Tuning Webcast replays at:
http://advancedatatools.com/Informix/Webcasts.html

Call: (800) 807-6732 x101 or Email: info@advancedatatools.com
Web: http://www.advancedatatools.com

Advanced DataTools

Thank You

Mike Walker

Advanced DataTools Corporation

mike@advancedatatools.com

For more information:

http://www.advancedatatools.com