Inside The OpenEdge RDBMS: After-Image Records and Their Uses

research from the parmington foundation

Gus Björklund

the parmington foundation

head groundskeeper

tpf

PUG CHALLENGE AMERICAS

Burlington, MA, USA
12 – 15 nov 2023

Notice

Please interrupt if you have a question.

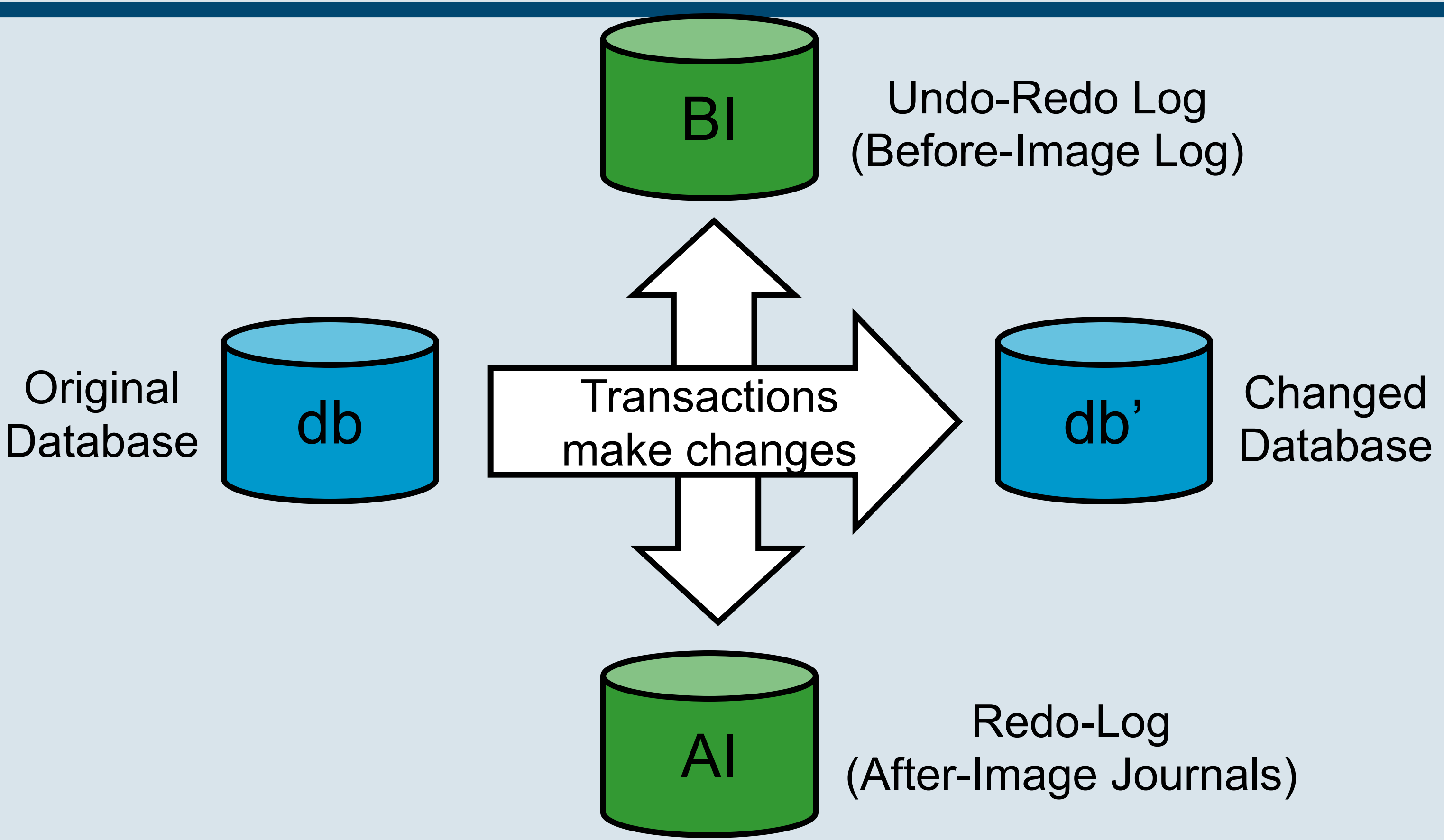
Sometimes we think that we are making sense when actually we are not.

who uses after-imaging ?

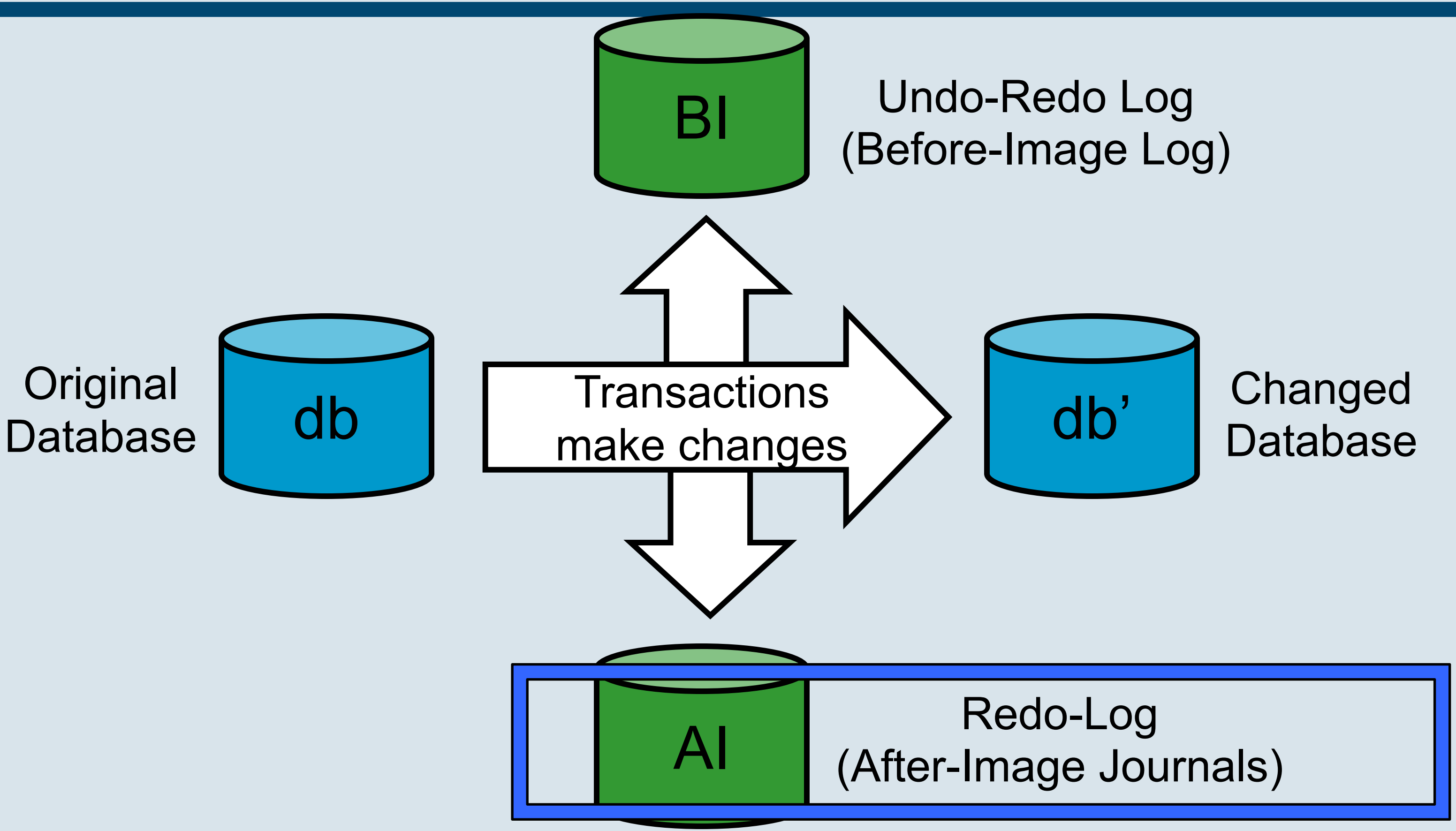
disaster strikes ! (ap photo)

After-image journaling to the rescue!

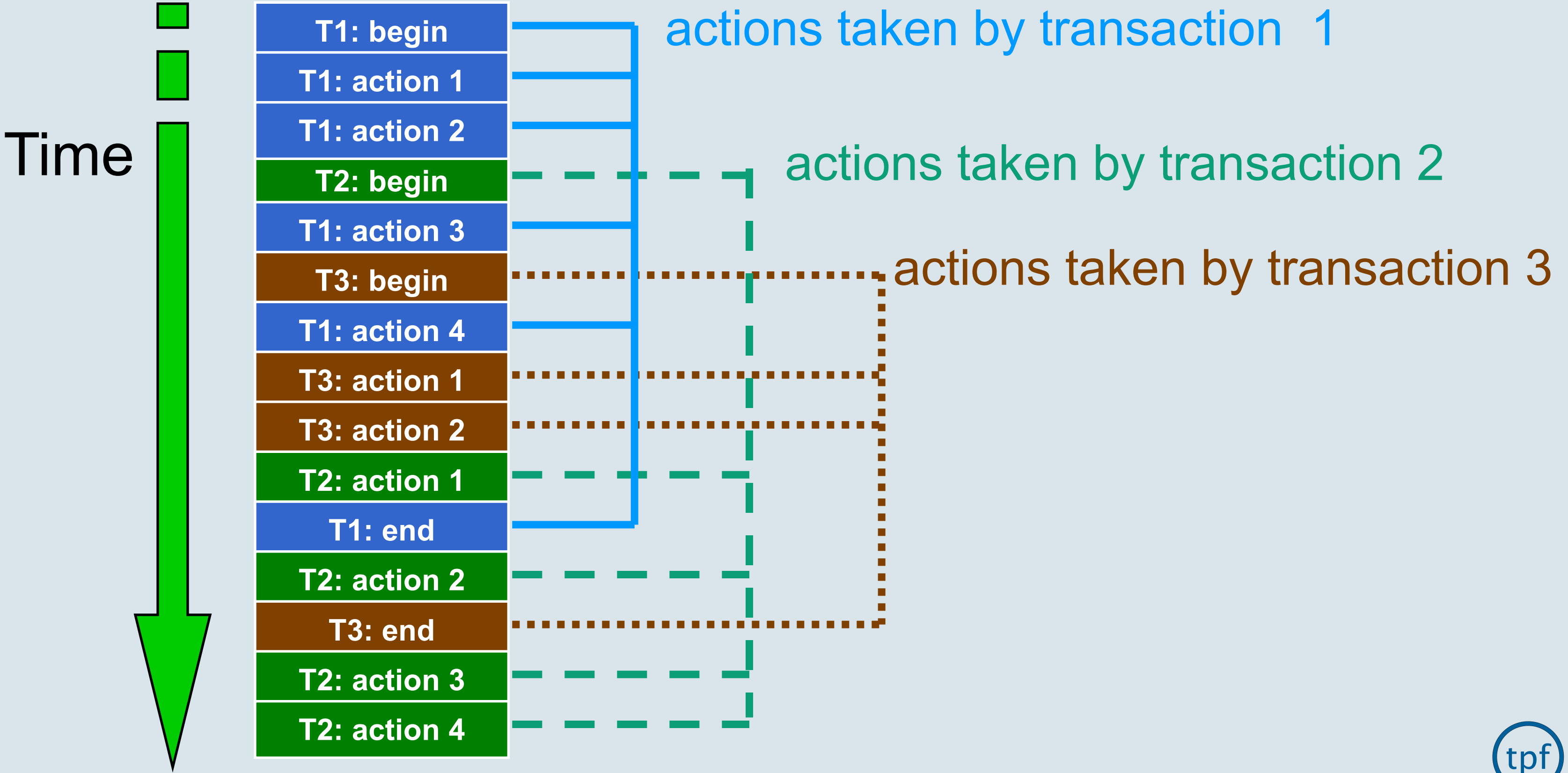
Two transaction logs



Two transaction logs



transaction log records (aka “notes”)



log records (notes)

- generated for every change to database.
- each describes exactly one change to one database block.
  - almost - there are log records that describe changes to purely memory-resident data structures like the transaction table
- apply only to specific version number of block
- some operations require more than one change
  - index splits, multi-block records
- written in same order changes are executed.
- notes from concurrent transactions are mixed together.

Undo-Redo (aka Before-Image) Log Records Contain

- transaction number
- data area number
- database block number (its dbkey)
- database block's version number
- note type – specifies what operation to perform

and . . .

Undo-Redo (aka Before-Image) Log Records Contain

- any information needed to undo the operation
  - in case we have to roll back
- any information needed to redo the operation
  - in case we lose the result before writing to disk
- any information needed to replicate the operation
  - duplicate the database at another server
  - no note, no replica

Redo (aka After-Image) Log Records Contain

- transaction number
- data area number
- database block number (its dbkey)
- database block's version number
- note type – specifies what operation to perform

and . . .

Redo (aka After-Image) Log Records Contain

- any information needed to undo the operation
  - in case we have to roll back
- any information needed to redo the operation
  - in case we lose the result before writing to disk
- any information needed to replicate the operation
  - duplicate the database at another server
  - no note, no replica

Did you notice anything about those two kinds of notes (bi and ai) ????

The So-Called "Before-Image" File Is a lie.

- Does not contain "before images"
- It has a record of all *recent* database changes
- The data are whatever is needed to:
  - Undo or roll back transactions
  - Perform crash recovery
- What is needed depends on the specific operation
  - Row create "after" row contents – "before" was nothing
  - Row delete has *current* row contents

The So-Called "After-Image" File Is a lie.

- Does not contain "after images"
- It has a record of *all* database changes after a full backup
  - call this point "time 0"
- The data are sufficient to:
  - Recover or recreate everything that happened since time 0
  - Recover or recreate the before-image log
  - Undo or roll back transactions
  - Perform crash recovery
  - Replicate the database in real-time (or later)

We need the logs because

- What if someone unplugs server to plug in vacuum cleaner?
- What if we want to undo (roll back) a typical transaction?
- What if we make several more changes and only one block of a fragmented record chain is written to disk to make room in the buffer pool ?
- What if DBA deletes entire database?

We need these because

- What if someone unplugs server to plug in vacuum cleaner?
  - the change will be lost
- What if we want to undo (rollback) ?
  - we don’t know the old value or how to undo
- What if we make several more changes and only one block of a fragmented record chain is written to disk to make room in the buffer pool ?
  - the database will be corrupted
- What if our ship runs aground and falls over
  - the database will disappear completely

these are all bad things (tm)

We know what’s in them.

Scanning the After-Image Journals

What can we do with that information?

Questions

- When was an ‘evil’ transaction started?
  Then i know how far to Roll Forward
- What transactions were active during an online backup?
  those would be reversed if the backup was restored
- What transactions were active during a quiet point?
  those would be reversed if the backup was restored
- What userid is causing the BI file to grow?
  and why?
- Is my database code crappy?
  Yes !
- Why did lock table overflow?
  Who caused it?

rfutil aimage scan reads an ai extent and reports

Prerequisites:

- After Imaging is enabled
- Use the "verbose" option to get exquisite detail
- aimage scan verbose is VERY verbose.
  you might use up all of your disk space and cause a disaster
- Prior to V10.1A the transaction index number (shown as *Trid* in the output) rolls over at 32767 in the report

How to do it

```
echo a foo.a1 >addai.st
prostrct add foo addai.st
rfutil foo -C mark
rfutil foo -C aimage begin
```

.... change stuff in database

```
rfutil bar -C aimage scan verbose \
       -a foo.a1 \
       > aireport.txt
```

NOTE: cannot do this with live ai extents,
only archived ones or copies
dbname can’t be “foo”

what do you get? 1 of 2

```
After-image dates for this database: (1632)
    Last AIMAGE BEGIN Thu Sep  7 14:28:36 2023 (1640)
    This is aimage file number 1 since the last AIMAGE BEGIN. (1642)
    This file was last opened for output on Thu Sep  7 14:28:36
2023.
After-image dates for this after-image file: (1633)
    Last AIMAGE BEGIN Thu Sep  7 14:28:36 2023 (1640)
    This is aimage file number 1 since the last AIMAGE BEGIN. (1642)
    This file was last opened for output on Thu Sep  7 14:28:36
2023.
```

what do you get? 2 of 2

```
Trid: 0 code = RL_INMEM version = 3 (12528)
Trid: 0 dbkey = 0  update counter = 0 (12530)
Trid: 0 code = RL_LSTMOD version = 2 (12528)
Trid: 0 area = 6   dbkey = 32   update counter = 13 (12529)
2 notes were processed. (1634)
0 in-flight transactions. (3785)
0 transactions were started. (1635)
0 transactions were completed. (11138)
At the end of the .ai file, 0 transactions were still active. (1636)
```

Selected note types and their meanings

Type	Meaning	Type	Meaning
RL_TBGN	Begin transaction	RL_SEINC	Next sequence value
RL_INMEM	Current transaction table	RL_CXINS	Insert index entry
RL_LSTMOD	Set database modified time	RL_CXREM	Remove index entry
RL_RMCR	Create record or fragment	RL_BKHWM	Set high water mark
RL_RMDEL	Delete record or fragment	BK_REPL	Byte string replace
RL_RMCHG	Change record’s contents	RL_TEND	End transaction

Now, let's examine some 4GL codes and the notes they produce

a few are from George Potemkin who gave us the idea for this
Спасибо !

Shell script

```
prodb s2000 sports2000
echo a s2000.a1 >addai.st
prostrct add s2000 addai.st
#
i=0
while [ ${i} -le 16 ]
do
    echo ”doing transaction ${i}”
    probkup s2000 /dev/null
    rfutil s2000 -C aimage begin
    pro s2000 -p t${i}.p
    rfutil s2000 -a s2000.a1 -C aimage scan verbose >scan${i}.txt
    rfutil s2000 -C aimage end
    i=`expr ${i} + 1`
done
```

first, we play with the customer table

example 0

```
DO TRANSACTION:
  FIND Customer 1 NO-LOCK NO-ERROR.
END.
```

example 1

```
DO TRANSACTION:
  FIND Customer 1 EXCLUSIVE-LOCK NO-ERROR.
END.
```

example 2

```
DO TRANSACTION:
  CREATE Customer.
  DELETE Customer.
END.
```

```
Trid: 27956 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27956 User Id: gus (12531)
Trid: 27956 code = RL_TBGN version = 1 (12528)
Trid: 27956 dbkey = 0  update counter = 0 (12530)
Trid: 27956 code = RL_SEINC version = 1 (12528)
Trid: 27956 area = 6   dbkey = 96   update counter = 27 (12529)
Trid: 27956 code = RL_TMSAVE version = 4 (12528)
Trid: 27956 dbkey = 0  update counter = 0 (12530)
Trid: 27956 code = RL_RMCR version = 3 (12528)
Trid: 27956 area = 9   dbkey = 1344   update counter = 22 (12529)
Trid: 27956 code = RL_CXINS version = 2 (12528)
Trid: 27956 area = 10   dbkey = 640   update counter = 220 (12529)
Trid: 27956 code = RL_CXREM version = 2 (12528)
Trid: 27956 area = 10   dbkey = 640   update counter = 221 (12529)
Trid: 27956 code = RL_RMDEL version = 3 (12528)
Trid: 27956 area = 9   dbkey = 1344   update counter = 23 (12529)
Trid: 27956 code = RL_RMCR version = 3 (12528)
Trid: 27956 area = 9   dbkey = 1344   update counter = 24 (12529)
Trid: 27956 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27956 code = RL_TEND version = 1 (12528)
Trid: 27956 dbkey = 0  update counter = 0 (12530)
```

example 3

```
DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  CREATE Customer.
  DELETE Customer.
END.
```

example 4

```
DEF VAR N AS INT NO-UNDO.

DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  CREATE Customer.
  N = RECID(Customer).
  DELETE Customer.
END.
```

```
Trid: 27958 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27958 User Id: gus (12531)
Trid: 27958 code = RL_TBGN version = 1 (12528)
Trid: 27958 dbkey = 0  update counter = 0 (12530)
Trid: 27958 code = RL_RMDEL version = 3 (12528)
Trid: 27958 area = 9   dbkey = 1344   update counter = 25 (12529)
Trid: 27958 code = RL_RMCR version = 3 (12528)
Trid: 27958 area = 9   dbkey = 1344   update counter = 26 (12529)
Trid: 27958 code = RL_RMDEL version = 3 (12528)
Trid: 27958 area = 9   dbkey = 1344   update counter = 27 (12529)
Trid: 27958 code = RL_RMCR version = 3 (12528)
Trid: 27958 area = 9   dbkey = 1344   update counter = 28 (12529)
Trid: 27958 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27958 code = RL_TEND version = 1 (12528)
Trid: 27958 dbkey = 0  update counter = 0 (12530)
```

next, we do stuff with the order table

example 5

```
create order.
assign
    custnum = 1
    orderdate = today
    ordernum = time
    shipdate = today + 1
    .
```

```
Trid: 27959 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27959 User Id: gus (12531)
Trid: 27959 code = RL_TBGN version = 1 (12528)
Trid: 27959 dbkey = 0  update counter = 0 (12530)
Trid: 27959 code = RL_SEINC version = 1 (12528)
Trid: 27959 area = 6   dbkey = 96   update counter = 28 (12529)
Trid: 27959 code = RL_TMSAVE version = 4 (12528)
Trid: 27959 dbkey = 0  update counter = 0 (12530)
Trid: 27959 code = RL_RMCR version = 3 (12528)
Trid: 27959 area = 11   dbkey = 20544   update counter = 14 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 4768   update counter = 354 (12529)
```

part 1 ...

```
Trid: 27959 code = RL_CXREM version = 2 (12528)
Trid: 27959 area = 11   dbkey = 4768   update counter = 355 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 5024   update counter = 116 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 4768   update counter = 356 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 3104   update counter = 474 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 384   update counter = 3956 (12529)
Trid: 27959 code = RL_CXINS version = 2 (12528)
Trid: 27959 area = 11   dbkey = 4704   update counter = 215 (12529)
Trid: 27959 code = RL_RMCHG version = 3 (12528)
Trid: 27959 area = 11   dbkey = 20544   update counter = 15 (12529)
Trid: 27959 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27959 code = RL_TEND version = 1 (12528)
Trid: 27959 dbkey = 0  update counter = 0 (12530)
```

part 2

example 6

```
disable triggers for load of order.

create order.
assign
    custnum = 1
    orderdate = today
    ordernum = time
    shipdate = today + 1
    .
```

```
Trid: 27960 code = RL_TBGN version = 1 (12528)
Trid: 27960 dbkey = 0  update counter = 0 (12530)
Trid: 27960 code = RL_RMCR version = 3 (12528)
Trid: 27960 area = 11   dbkey = 20544   update counter = 16 (12529)
Trid: 27960 code = RL_CXINS version = 2 (12528)
Trid: 27960 area = 11   dbkey = 5024   update counter = 117 (12529)
Trid: 27960 code = RL_CXINS version = 2 (12528)
Trid: 27960 area = 11   dbkey = 4768   update counter = 357 (12529)
Trid: 27960 code = RL_CXINS version = 2 (12528)
Trid: 27960 area = 11   dbkey = 3104   update counter = 475 (12529)
Trid: 27960 code = RL_CXINS version = 2 (12528)
Trid: 27960 area = 11   dbkey = 384   update counter = 3957 (12529)
Trid: 27960 code = RL_CXINS version = 2 (12528)
Trid: 27960 area = 11   dbkey = 4704   update counter = 216 (12529)
Trid: 27960 Thu Sep  7 14:28:36 2023. (2598)
Trid: 27960 code = RL_TEND version = 1 (12528)
Trid: 27960 dbkey = 0  update counter = 0 (12530)
```

example 7

```
disable triggers for load of order.

create order.
assign
    custnum = 1
    orderdate = today
    ordernum = time.
    shipdate = today + 1
    .
```

```
Trid: 27961 code = RL_RMCR version = 3 (12528)
Trid: 27961 area = 11   dbkey = 20544   update counter = 17 (12529)
Trid: 27961 code = RL_CXINS version = 2 (12528)
Trid: 27961 area = 11   dbkey = 864   update counter = 68 (12529)
Trid: 27961 code = RL_CXINS version = 2 (12528)
Trid: 27961 area = 11   dbkey = 4768   update counter = 358 (12529)
Trid: 27961 code = RL_CXINS version = 2 (12528)
Trid: 27961 area = 11   dbkey = 3104   update counter = 476 (12529)
Trid: 27961 code = RL_CXINS version = 2 (12528)
Trid: 27961 area = 11   dbkey = 384   update counter = 3958 (12529)
Trid: 27961 code = RL_CXINS version = 2 (12528)
Trid: 27961 area = 11   dbkey = 4704   update counter = 217 (12529)
Trid: 27961 code = RL_RMCHG version = 3 (12528)
Trid: 27961 area = 11   dbkey = 20544   update counter = 18 (12529)
```

example 8 (... serendipity)

```
find order 1.
information = fill("abc", 6000).
```

```
Trid: 27962 code = RL_RMDEL version = 3 (12528)
Trid: 27962 area = 11   dbkey = 96   update counter = 39 (12529)
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 96   update counter = 40 (12529)
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 20544   update counter = 19 (12529)
Trid: 27962 code = RL_BKFRM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1767 (12529)
Trid: 27962 code = RL_BKFRB version = 2 (12528)
Trid: 27962 area = 11   dbkey = 20544   update counter = 20 (12529)
Trid: 27962 code = RL_BKMAKE version = 1 (12528)
Trid: 27962 area = 11   dbkey = 20576   update counter = 0 (12529)
Trid: 27962 code = RL_BKHWM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1768 (12529)
```

```
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 20576   update counter = 1 (12529)
Trid: 27962 code = RL_BKMAKE version = 1 (12528)
Trid: 27962 area = 11   dbkey = 20608   update counter = 0 (12529)
Trid: 27962 code = RL_BKHWM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1769 (12529)
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 20608   update counter = 1 (12529)
Trid: 27962 code = RL_BKMAKE version = 1 (12528)
Trid: 27962 area = 11   dbkey = 20640   update counter = 0 (12529)
Trid: 27962 code = RL_BKHWM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1770 (12529)
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 20640   update counter = 1 (12529)
          .......
```

```
Trid: 27962 code = RL_BKMAKE version = 1 (12528)
Trid: 27962 area = 11   dbkey = 20672   update counter = 0 (12529)
Trid: 27962 code = RL_BKHWM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1771 (12529)
Trid: 27962 code = RL_RMCR version = 3 (12528)
Trid: 27962 area = 11   dbkey = 20672   update counter = 1 (12529)
Trid: 27962 code = RL_BKFAB version = 1 (12528)
Trid: 27962 area = 11   dbkey = 20672   update counter = 2 (12529)
Trid: 27962 code = RL_BKFAM version = 2 (12528)
Trid: 27962 area = 11   dbkey = 64   update counter = 1772 (12529)
Trid: 27962 code = RL_RMNXTF version = 2 (12528)
Trid: 27962 area = 11   dbkey = 20640   update counter = 2 (12529)
Trid: 27962 code = RL_RMNXTF version = 2 (12528)
Trid: 27962 area = 11   dbkey = 20608   update counter = 2 (12529)
Trid: 27962 code = RL_RMNXTF version = 2 (12528)
Trid: 27962 area = 11   dbkey = 20576   update counter = 2 (12529)
Trid: 27962 code = RL_RMNXTF version = 2 (12528)
Trid: 27962 area = 11   dbkey = 20544   update counter = 21 (12529)
Trid: 27962 code = RL_RMNXTF version = 2 (12528)
```

now for some other stuff

example 9

```
DO TRANSACTION:
  CREATE _View.
  DELETE _View.
END.
```

```
Trid: 27963 Thu Sep  7 14:28:37 2023. (2598)
Trid: 27963 User Id: gus (12531)
Trid: 27963 code = RL_TBGN version = 1 (12528)
Trid: 27963 dbkey = 0  update counter = 0 (12530)
Trid: 27963 code = RL_RMCR version = 3 (12528)
Trid: 27963 area = 6   dbkey = 14144   update counter = 7 (12529)
Trid: 27963 code = RL_RMDEL version = 3 (12528)
Trid: 27963 area = 6   dbkey = 14144   update counter = 8 (12529)
Trid: 27963 code = RL_RMCR version = 3 (12528)
Trid: 27963 area = 6   dbkey = 14144   update counter = 9 (12529)
Trid: 27963 Thu Sep  7 14:28:37 2023. (2598)
Trid: 27963 code = RL_TEND version = 1 (12528)
Trid: 27963 dbkey = 0  update counter = 0 (12530)
```

example 10

```
DEF VAR I AS INT.

DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  DO i = 1 TO 1:
    CREATE Customer.
    DELETE Customer.
  END.
END.
```

example 11

```
DEF VAR I AS INT.

DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  REPEAT i = 1 TO 1:
    CREATE Customer.
    DELETE Customer.
  END.
END.
```

```
Trid: 27965 Thu Sep  7 14:28:37 2023. (2598)
Trid: 27965 User Id: gus (12531)
Trid: 27965 code = RL_TBGN version = 1 (12528)
Trid: 27965 dbkey = 0  update counter = 0 (12530)
Trid: 27965 code = RL_TMSAVE version = 4 (12528)
Trid: 27965 dbkey = 0  update counter = 0 (12530)
Trid: 27965 Thu Sep  7 14:28:37 2023. (2598)
Trid: 27965 code = RL_TEND version = 1 (12528)
Trid: 27965 dbkey = 0  update counter = 0 (12530)
```

subtransaction causes transaction to start
0 notes if using -nosavepoint

example 12

```
DEF VAR N AS INT NO-UNDO.

DO TRANSACTION:
  N = CURRENT-VALUE(NextCustNum).
END.
```

example 13

```
DEF VAR N AS INT NO-UNDO.

DO TRANSACTION:
  N = NEXT-VALUE(NextCustNum).
END.
```

```
Trid: 27967 Thu Sep  7 14:28:38 2023. (2598)
Trid: 27967 User Id: gus (12531)
Trid: 27967 code = RL_TBGN version = 1 (12528)
Trid: 27967 dbkey = 0  update counter = 0 (12530)
Trid: 27967 code = RL_SEINC version = 1 (12528)
Trid: 27967 area = 6   dbkey = 96   update counter = 29 (12529)
Trid: 27967 Thu Sep  7 14:28:38 2023. (2598)
Trid: 27967 code = RL_TEND version = 1 (12528)
Trid: 27967 dbkey = 0  update counter = 0 (12530)
```

example 13.1

```
DEF VAR N AS INT NO-UNDO.

N = NEXT-VALUE(NextCustNum).
```

```
Trid: 0 code = RL_SEINC version = 1 (12528)
Trid: 0 area = 6   dbkey = 96   update counter = 30 (12529)
```

Why are there no TBGN and TEND notes?

Why indeed?

Not all notes need to be “inside” a transaction

- Sequence incrementing
- Extent switches
- Expanding the size of variable extents
- Transaction Begin
- Etc .

example 14

```
DEF VAR I AS INT.
DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  REPEAT i = 1 TO 1:
    CREATE Customer.
    UNDO.
  END.
END.
```

```
Trid: 27968 Thu Sep  7 14:28:38 2023. (2598)
Trid: 27968 User Id: gus (12531)
Trid: 27968 code = RL_TBGN version = 1 (12528)
Trid: 27968 dbkey = 0  update counter = 0 (12530)
Trid: 27968 code = RL_TMSAVE version = 4 (12528)
Trid: 27968 dbkey = 0  update counter = 0 (12530)
Trid: 27968 code = RL_TMSAVE version = 4 (12528)
Trid: 27968 dbkey = 0  update counter = 0 (12530)
Trid: 27968 Thu Sep  7 14:28:38 2023. (2598)
Trid: 27968 code = RL_TEND version = 1 (12528)
Trid: 27968 dbkey = 0  update counter = 0 (12530)
```

subtransactions cause main transaction to start
0 notes if using -nosavepoint

example 15

```
DISABLE TRIGGERS FOR LOAD OF Customer.

DO TRANSACTION:
  CREATE Customer.
  UNDO.
END.
```

example 16

```
DO TRANSACTION:
  FIND FIRST Customer NO-LOCK
    WHERE RECID(Customer) EQ 1 NO-ERROR.
  UNDO.
END.
```

example 17

```
do transaction:
    find _myconnect.
    _myconn-numseqbuffers = 3.
end.
```

So ...

- When was an ‘evil’ transaction started?
- What transactions were active during an online backup?
- What transactions were active during a quiet point?
- What userid is causing the BI file to grow? and why?
- Is my database code crappy?
- Why did lock table overflow? Who caused it?
- Why do the AI files grow so fast?

go home and try it !

you'll like it.

C
COSTA CONCORDIA
COSTA CONCORDIA

Questions?

email:

gus@parmington.com

Research from the parmington foundation