

# Obsolete DBA Best Practices



Nectar Daloglou, White Star Software  
[nd@wss.com](mailto:nd@wss.com)

White Star Software

# A Few Words about the Speaker



- 
- Nectar Daloglou; Progress & QAD since 2000.
  - Performed specialized services at more than 60 Progress customer sites:
    - Progress Database Administration
    - Install/Upgrades/Migrations of Progress and QAD Applications
    - Technical Audits / Performance Tuning
    - Business Continuity Strategies
  - Now part of White Star Software
  - Key Member in large Progress and QAD implementation

# Agenda

---



- Introduction
- Improve Performance
- Save Time
- Reduce Downtime
- Other Interesting Tricks
- Question

# Before We Start...



- 
- *Your Mileage May Vary*
    - Information in this presentation may not necessarily apply to your environment
    - Verify your best practices with an expert
    - The recommendations are general and apply mostly to OpenEdge 10+ environments

# Is your Progress version obsolete?



Product Version & Current Service Pack (SP)	Life Cycle by Product Release		
	Active	Mature	Retired
OpenEdge 11.7.0	2017-Mar	-	-
OpenEdge 11.6.3	2015-Oct	2017-Mar	2019-Mar (T)
OpenEdge 11.5.1	2014-Dec	2015-Oct	2017-Dec (T)
OpenEdge 11.4.0	2014-Aug	2014-Dec	2017-Aug (T)
OpenEdge 11.3.3	2013-Jul	2014-Aug	2016-Aug
OpenEdge 11.2.1	2013-Feb	2013-Jul	2016-Feb
OpenEdge 11.1.0	2012-Jun	2013-Feb	2014-Feb
OpenEdge 11.0.0	2011-Dec	2012-Jun	2013-Jun
OpenEdge 10.2B.08	2009-Dec	2011-Dec	tbd
OpenEdge 10.2A	2008-Nov	2009-Dec	2010-Feb
OpenEdge 10.1C	2008-Feb	2009-Aug	2014-Jul
OpenEdge 10.1B	2006-Dec	2008-May	2008-May
OpenEdge 10.1A	2005-Dec	2007-Mar	2007-Mar
OpenEdge 10.0B	2004-Aug	2006-Mar	2006-Mar
OpenEdge 10.0A	2003-Dec	2004-Nov	2004-Nov
Progress V9.1E	2004-Nov	2006-Nov	2015-Oct
Progress V9.1D	2002-May	2005-Feb	2005-Feb
Progress V8.3E	2001-Dec	2002-Mar	2010-Feb
Progress V8.3D	2000-Jan	2002-Mar	2002-Mar

White Star Software

(T) indicates the Target date for moving to the Retired Phase. **Reference:**

[https://community.progress.com/community\\_groups/openedge\\_general/w/openedgegeneral/1501.openedge-product-availability-guides-and-life-cycle-guide - June-2017](https://community.progress.com/community_groups/openedge_general/w/openedgegeneral/1501.openedge-product-availability-guides-and-life-cycle-guide - June-2017)

# Product Life Cycle In Brief

---



- **Active:**
  - Fully supported and receives new features and enhancements.
- **Mature:**
  - Evaluated for certifications on new operating environments but not new hardware systems

Reference: Progress OpenEdge Life Cycle Policy Guide - July 2016 Update

# Product Life Cycle In Brief

---



- Retired:
  - Not available for sale, only for adding user counts to existing licenses;
  - Not evaluated for certifications on new operating environments;
  - No Service Packs, nor Hot Fixes

# QUESTION



**Which of the following obsolete items below can still be useful?**





# ANSWER



**All of the items below are considered obsolete but can still be used for one reason or another.**



# Still useful...



White Star Software

# Obsolete Practices



- Common Reasons
  - *If it ain't broke why fix it?*
    - Reduced Performance
    - Longer to maintain than required
    - Unnecessary downtime
  - Version or system does not support the new method
  - New functionalities are not known
  - The practices have not been revised since the upgrade

# Agenda



- 
- Introduction
  - Improve Performance
  - Save Time
  - Reduce Downtime
  - Other Interesting Tricks
  - Questions



# Obsolete Practice

---

- Regularly Truncating the BI
  - *proutil <DB> -C truncate bi*



# Regularly Truncating the BI



- 
- Common Reasons:
    - Save space on the disk drives or backup tape
    - Avoid an uncontrollable size
    - Prevent data-loss
    - Monitor BI growth

# BI In Brief



- 
- Every transaction is logged in the BI file;
  - Ensures database integrity:
    - Undoes or reverses uncommitted transactions in the event of a failure or crash;
    - Restores the DB to a consistent state after a failure or crash



# BI In Brief (con't)

---

- BI space is re-used when it's no longer needed
- The BI consists of clusters
  - Default 512KB
- All transactions freeze when the BI is growing and formatting



Reference: Progress Kbase #13866 – Undo and Redo Logging: How the BI file works



# BI Formatting



White Star Software

# Solution

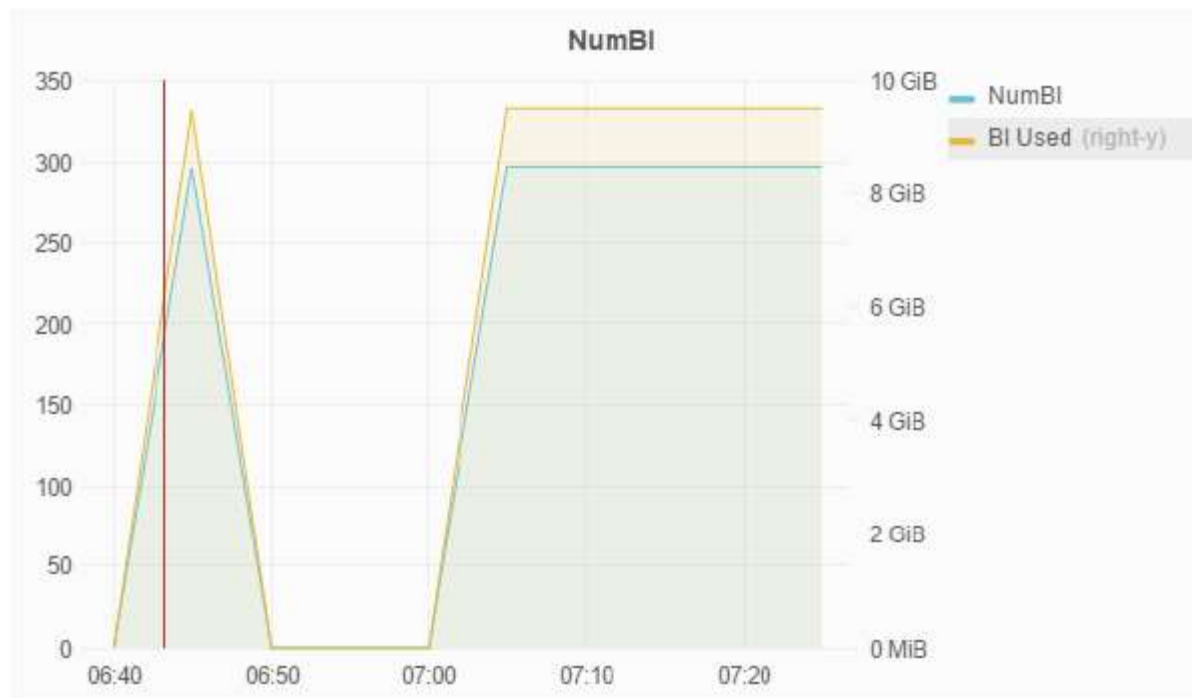


- Avoid truncating the BI
- Use BI Grow:
  - » *proutil <DB> -C bigrow <# clusters>*
- Keep track of BI file growth
  - Identify the BI’s “natural size”
  - Can be tracked using the VST:
    - for each \_AreaStatus where \_AreaStatus-Areanum = 3:*
    - disp \_AreaStatus-Totblocks*
  - Or by the BI file itself if using a variable extent:
    - ls -l qaddb.bi*
    - rw-r--r-- 1 mfg qad 1148276008 Sep 18 00:17 qaddb.bi*

# Monitor BI with ProTop



- View BI usage
- Set Thresholds and receive alert (Paid version)





# Valid Truncate Reasons

---

- Valid Reasons for Truncating:
  - Reduce the size after an unusual operation.
    - Example:
      - Dump/Load
      - After a very largely scoped or old transaction has occurred
  - Access a database in Read-Only mode:

```
----- Error -----  
WARNING: Before-image file of database /reg03/dbs/hlp/r03prd1hlp is not  
truncated. (1552)
```

```
-----  
<OK>
```

# Valid Truncate Reasons

---



- Valid Reasons for Truncating (con't):
  - When the truncate is followed by the “BIGROW” command;
  - Progress automatically truncates during certain operations:
    - Example:
      - When activating After-Imaging
      - When changing the BI block size or cluster size



# Obsolete Practice

---

- Using your database like a version 8 DB
  - Using the default “Schema Area”

```
ADD TABLE "Acc_bal"  
AREA "Schema Area"  
DESCRIPTION "Account Balances"
```

```
ADD INDEX "Key" ON "Acc_bal"  
AREA "Schema Area"  
UNIQUE  
PRIMARY
```

- Using Type 1 Storage Areas:

```
d "TRANSACTION":7,64; I  
d "TRHIST":25,64; I
```



# Storage Areas In Brief

---

- Three types: “0”, 1 and 2
  - ALWAYS use Type 2
- Define
  - Where to store database objects;
  - Where to place the actual database files;
  - Records per block
  - Blocks per cluster

# “Schema Area” Storage Area



- Contains Master Block and other DB structure info;
- Cannot change
  - RPB=32 and BPC=1 (Type 1)
  - *d “Schema Area”:6,32;1*
- Cannot physically segregate DB objects







# Type 2 Storage Areas

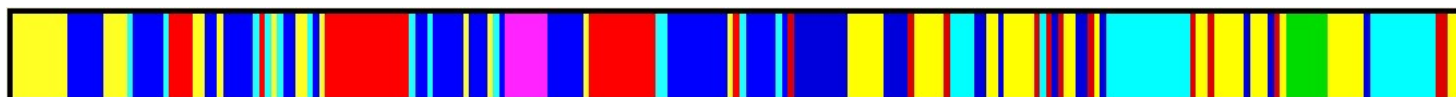
---

- Defined at DB creation
- BPC > 1
  - *d* “TRANSACTION”:7,128;512
  - *d* “STATIC”:9,64;8
- Store database objects in clusters;
- Reduces scatter
  - Object blocks are contiguous



# Type 2 Storage Areas

- TRANSACTION AREA in Type I



- TRANSACTION AREA in Type II



Abs\_mstr



Spt\_det



Txnd\_det



Sch\_mstr



Wod\_det



Opgl\_det

# Agenda



- 
- Introduction
  - Improve Performance
  - Save Time
  - Reduce Downtime
  - Other Interesting Tricks
  - Questions

# Obsolete Practice



- 
- Scripts for managing After-Imaging
    - After-Imaging is essential for restoring a DB
    - Without AI, we can only restore from our last good backup
    - Until version 10.1A, managing After-Imaging has been a challenge.



# After-Imaging Scripts

- AI once required complex scripting

```
#!/usr/bin/ksh## Script to check AI extents and switch, if necessary.# Set common variables. /dba/scripts/setenv.qaddbTHISSCRIPT=${SCRIPTNAME}=`basename
THISSCRIPT`usage="Usage: $0 <DB> <FORCE>"BACKDIR=/work/PRD/qaddbbkup; export BACKDIRALTPATH=noneFORCE=noFAILED=0# Set up error notification e-mail
headerEMAILSUBJ="`hostname`: Error in $THISSCRIPT"EMAILTO="dba@dba.com"failcheck(){if [ $? -ne 0 ];then echo "" echo "***** Script Failed Here *****" echo ""
let FAILED+=1fi)if [ $# = 1 ] || [ $# = 2 ];then DB=$1 DBNAME=`basename $DB` # Test if DB exists and if enabled for AI and valid # with Progress version _rfutil $DB-
C aimage extent list > /tmp/tmpextlist$$ if [ $? = 0 ];then continue else echo echo "ERROR! $DB is not a valid DB for AI or Progress version not
compatible" echo Exiting... rm /tmp/tmpextlist$$ exit 1 exit 1 fi24%0 # Test if FORCE parameter is specified if [ $# = 2 ];then if [
$2 = "FORCE" ];then FORCE=yes else echo $2 is not a valid parameter echo $usage exit 1 fi else echo
echo $usage exit 1fi# Exit if the previous iteration of this script is sleeping.# Skip this check if FORCE parameter is
specifiedRUNNINGFLAG=/home/dbmanage/$SCRIPTNAME.$DBNAME.running;export RUNNINGFLAGif [ $FORCE = "no" ] && [ -f $RUNNINGFLAG ];then echo $THISSCRIPTis
already running exit%)0fiecho RUNNING FLAG: $RUNNINGFLAGtouch $RUNNINGFLAGchmod 640 $RUNNINGFLAGchown dbmanage $RUNNINGFLAGcat >> $RUNNINGFLAG <<
EOF`date`$THISSCRIPT $1EOFecho +-----+datelet NumExtents=`grep -c Extent: /tmp/tmpextlist$$`let FullExtents=`grep -c "Status:
Full" /tmp/tmpextlist$$`let LockedExtents=`grep -c "Status: Locked" /tmp/tmpextlist$$`let EmptyExtents=`grep -c "Status: Empty" /tmp/tmpextlist$$`let
MinEmptyExtents=$NumExtents-1let MinEmptyExtents=$LockedExtents#let MaxFullExtents=$NumExtents-2echo +-----+echo $DBecho "Number of
Extents: " $NumExtentsecho "Full Extents: " $FullExtentsecho "Locked Extents: " $LockedExtentsecho "Empty Extents: " $EmptyExtents# Backup all full extents and mark them as
empty, then switch AI. if [ $FullExtents -gt 0 ];then #echo "Maximum Full Extents: " $MaxFullExtents echo "Actual Full Extents: " $FullExtents echo "Actual Empty Extents: "
$EmptyExtents CurrFullExtent=`_rfutil $DB -C aimage extent full` if [ $? -eq 0 ]; then while [ $EmptyExtents -lt $MinEmptyExtents ];do
CurrFullExtent=`_rfutil $DB -C aimage extent full` BackupFileName=$BACKDIR/"`basename $CurrFullExtent`.date +%Y-%m-%d.%H.%M.%S` echo COPY
$CurrFullExtent to $BackupFileName cp $CurrFullExtent $BackupFileName failcheck gzip -v $BackupFileName failcheck if [ $FAILED = 0 ];
then _rfutil $DB -C aimage extent empty _rfutil $DB -C aimage extent list > /tmp/tmpnewlist$$ let EmptyExtents=`grep -c "Status: Empty"
/tmp/tmpnewlist$$` rm /tmp/tmpnewlist$$ else break fi done fi firm /tmp/tmpextlist$$# Execute RSYNC script in
background#/dba/scripts/rsync.sh >> /var/adm/rsync_log 2>/dev/null && # Remove running flag unless FORCE is specifiedif [ $FORCE = "no" ];then rm $RUNNINGFLAGfiif [
$FAILED -gt 0 ]; then date echo Error in AI Check Script cat > /tmp/msg$$ << EOFThere was an error in the AI check script for $DB .Please check the log file in
/tmp/$SCRIPTNAME.$DBNAME.log.EOF mailx -s "$EMAILSUBJ" $EMAILTO < /tmp/msg$$ rm /tmp/msg$$fiecho +-----+
```

# After-Image Scripting

---



- AI Management:
  - Avoid Downtime
    - Ensure empty extents are available
    - Ensure AI script is running
  - Be prepared in case of failure
    - Archive all full extents
    - Assign descriptive name to archived extents
    - Pair archive extents with database backup files

# AI Management Utility



- 
- A.K.A. AI File Management Tool, AI Archiver
  - Replaces most of the AI scripting
  - AI Archiver archives AI extents as soon as they become full or through a specified interval
  - Gives a descriptive yet long name to archived extent:

~aisave~v10~db1~00009.0001.db1.a1

# AI Management Utility



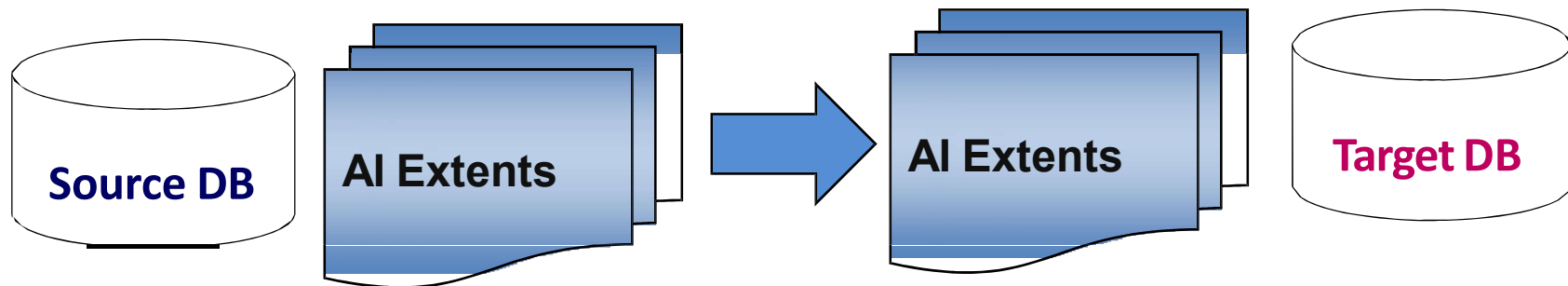
- Steps to enable:
  1. *rfutil <DB> -C aiarchiver enable*
  2. Add parameters in database startup script:  
*proserve <DB> [parameters]*  
*-aiarcdir <acrhive directory>*  
*-aiarcinterval <seconds>*
- Also possible to activate online with the probkup command:  
*probkup online <DB> <DB>.bak enableai enableaiarchiver -aiarcdir <dir>*





# Obsolete Practice

- After-Image File Replication or “Poor Man’s Replication”
  - Uses AI files to replicate to a secondary server for a warm spare or business continuity
  - Uses scripts to transfer full AI extents and roll them forward to a warm spare database



# AI Replication



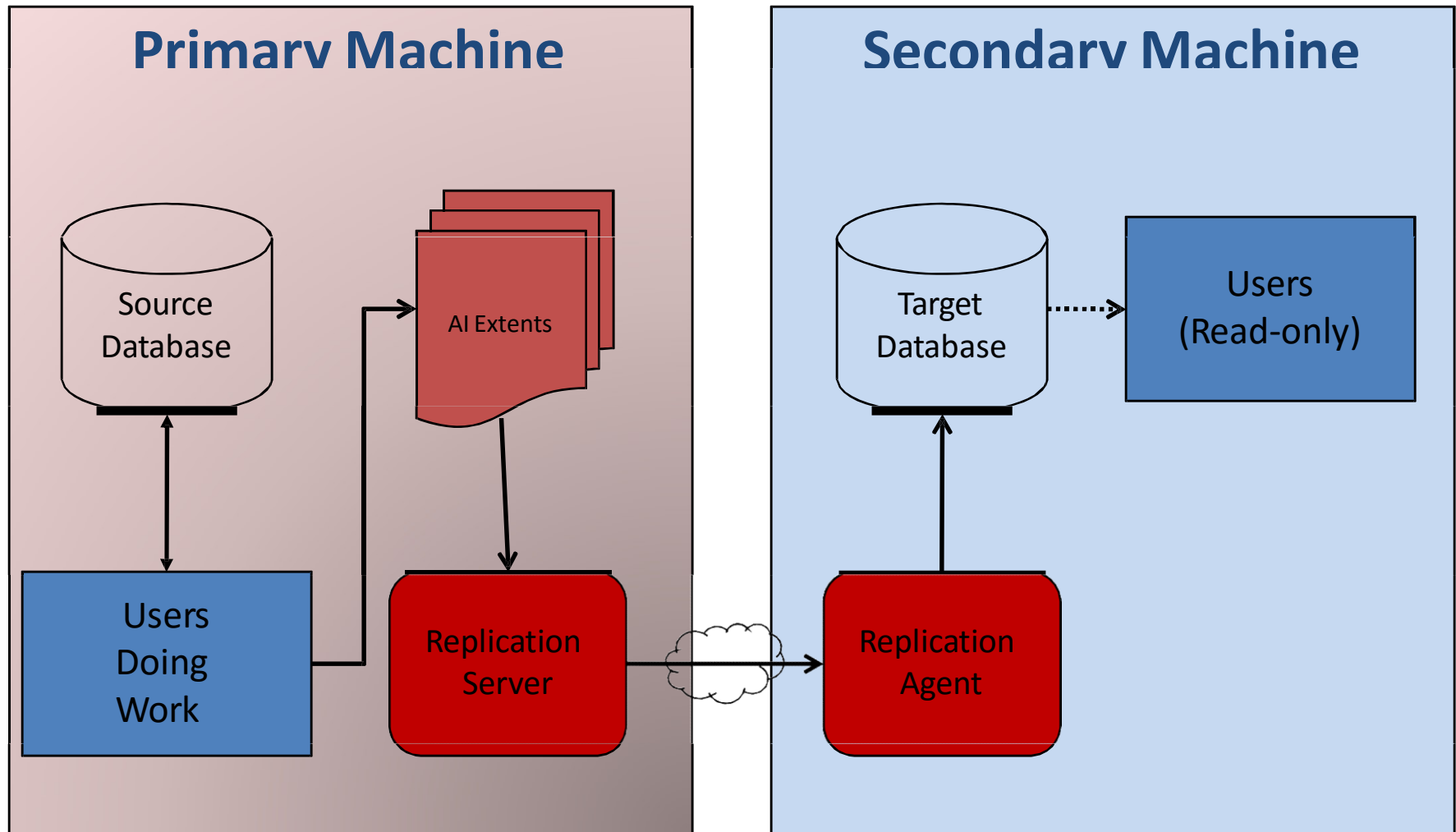
- 
- Requires complex scripting
  - Often requires minor adjustments and manual intervention
  - Still requires a Progress license on target server
  - The OpenEdge Replication license often costs less than the target server's duplicate set of licenses

# OpenEdge Replication



- 
- OpenEdge Replication helps in recovering from a disaster by replicating the Progress databases to another server or site
  - The target database can be used for reporting
  - Low latency

# OpenEdge Replication Architecture





# Obsolete Practice

---

- Over emphasizing fixed extents
  - d "TRANSACTION":7,64;8 f 200000
  - d "TRANSACTION":7,64;8
  - d "TRANSACTION\_IDX":8,64;8 f 20000
  - d "TRANSACTION\_IDX":8,64;8 f
  - d "STATIC":9,64;8 f 600000
  - d "STATIC\_IDX":9,64;8
  - d "HISTORY":9,64;8 f 200000
  - d "HISTORY\_IDX":9,64;8



# Fixed Extents

---

- Main Reason:
  - Writing in the variable extent may cause more I/O
- However...
  - Servers and disk subsystems are much faster than they were 10 years ago, especially the I/O performance
  - The performance penalty for variable extents is often negligible.



# Fixed Extents

---

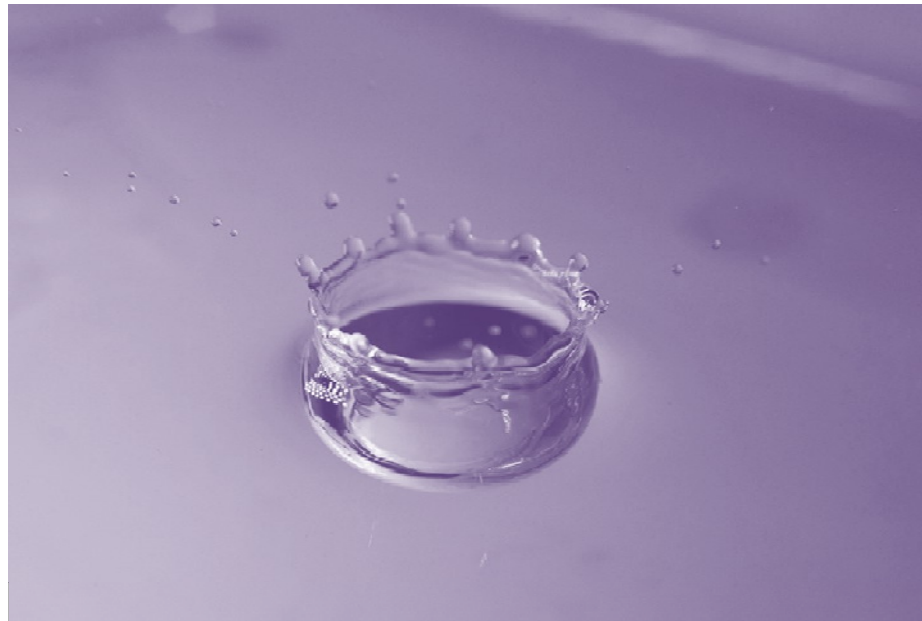
- Example for a Mon-Fri, 9/5 shop, growing 1 GB per month:
  - With a 4KB DB and 512 BPC, every extension is 2MB
  - $1\text{GB per month} / 2\text{ MB} = 500\text{ Database extensions}$
  - $500\text{ DB extensions} / 20\text{ working days} = 25\text{ extensions per day}$
  - $25\text{ extensions per day} / 8\text{ hours} = 3\text{ extensions per hour}$



# Fixed Extents

---

- Conclusion: Every 20 minutes, your users may have to wait a few milliseconds for your DB to grow 2 MB







# Obsolete Practice

- Maintaining DB extents less than 2GB:

qaddb.d1	qaddb_16.d3	qaddb_25.d18	qaddb_28.d2	qaddb_35.d12	qaddb_36.d12	qaddb_7.d1	qaddb_7.d7
qaddb.d2	qaddb_17.d1	qaddb_25.d19	qaddb_28.d3	qaddb_35.d13	qaddb_36.d13	qaddb_7.d10	qaddb_7.d8
qaddb.d3	qaddb_17.d2	qaddb_25.d2	qaddb_28.d4	qaddb_35.d14	qaddb_36.d14	qaddb_7.d11	qaddb_7.d9
qaddb.db	qaddb_18.d1	qaddb_25.d3	qaddb_28.d5	qaddb_35.d15	qaddb_36.d15	qaddb_7.d12	qaddb_8.d1
qaddb_10.d1	qaddb_18.d2	qaddb_25.d4	qaddb_28.d6	qaddb_35.d16	qaddb_36.d2	qaddb_7.d13	qaddb_8.d10
qaddb_10.d2	qaddb_19.d1	qaddb_25.d5	qaddb_28.d7	qaddb_35.d17	qaddb_36.d3	qaddb_7.d14	qaddb_8.d11
qaddb_10.d3	qaddb_19.d2	qaddb_25.d6	qaddb_29.d1	qaddb_35.d18	qaddb_36.d4	qaddb_7.d15	qaddb_8.d12
qaddb_11.d1	qaddb_19.d3	qaddb_25.d7	qaddb_29.d2	qaddb_35.d19	qaddb_36.d5	qaddb_7.d16	qaddb_8.d13
qaddb_11.d2	qaddb_20.d1	qaddb_25.d8	qaddb_29.d3	qaddb_35.d2	qaddb_36.d6	qaddb_7.d17	qaddb_8.d14
qaddb_11.d3	qaddb_20.d2	qaddb_25.d9	qaddb_30.d1	qaddb_35.d20	qaddb_36.d7	qaddb_7.d18	qaddb_8.d15
qaddb_11.d4	qaddb_20.d3	qaddb_26.d1	qaddb_30.d2	qaddb_35.d21	qaddb_36.d8	qaddb_7.d19	qaddb_8.d2
qaddb_11.d5	qaddb_21.d1	qaddb_26.d2	qaddb_30.d3	qaddb_35.d22	qaddb_36.d9	qaddb_7.d2	qaddb_8.d3
qaddb_12.d1	qaddb_21.d2	qaddb_26.d3	qaddb_31.d1	qaddb_35.d23	qaddb_37.d1	qaddb_7.d20	qaddb_8.d4
qaddb_12.d2	qaddb_22.d1	qaddb_26.d4	qaddb_31.d2	qaddb_35.d24	qaddb_37.d2	qaddb_7.d21	qaddb_8.d5
qaddb_12.d3	qaddb_22.d2	qaddb_26.d5	qaddb_31.d3	qaddb_35.d25	qaddb_38.d1	qaddb_7.d22	qaddb_8.d6
qaddb_13.d1	qaddb_23.d1	qaddb_26.d6	qaddb_32.d1	qaddb_35.d26	qaddb_38.d2	qaddb_7.d23	qaddb_8.d7
qaddb_13.d2	qaddb_23.d2	qaddb_26.d7	qaddb_32.d2	qaddb_35.d27	qaddb_39.d1	qaddb_7.d24	qaddb_8.d8
qaddb_13.d3	qaddb_24.d1	qaddb_27.d1	qaddb_32.d3	qaddb_35.d28	qaddb_39.d2	qaddb_7.d25	qaddb_8.d9
qaddb_14.d1	qaddb_24.d2	qaddb_27.d10	qaddb_32.d4	qaddb_35.d3	qaddb_39.d3	qaddb_7.d26	qaddb_9.d1
qaddb_14.d2	qaddb_25.d1	qaddb_27.d2	qaddb_33.d1	qaddb_35.d4	qaddb_39.d4	qaddb_7.d27	qaddb_9.d2
qaddb_15.d1	qaddb_25.d10	qaddb_27.d3	qaddb_33.d2	qaddb_35.d5	qaddb_39.d5	qaddb_7.d28	qaddb_9.d3
qaddb_15.d2	qaddb_25.d11	qaddb_27.d4	qaddb_33.d3	qaddb_35.d6	qaddb_40.d1	qaddb_7.d29	qaddb_9.d4
qaddb_15.d3	qaddb_25.d12	qaddb_27.d5	qaddb_34.d1	qaddb_35.d7	qaddb_40.d2	qaddb_7.d3	
qaddb_15.d4	qaddb_25.d13	qaddb_27.d6	qaddb_34.d2	qaddb_35.d8	qaddb_40.d3	qaddb_7.d30	
qaddb_15.d5	qaddb_25.d14	qaddb_27.d7	qaddb_34.d3	qaddb_35.d9	qaddb_41.d1	qaddb_7.d31	
qaddb_15.d6	qaddb_25.d15	qaddb_27.d8	qaddb_35.d1	qaddb_36.d1	qaddb_41.d2	qaddb_7.d4	
qaddb_16.d1	qaddb_25.d16	qaddb_27.d9	qaddb_35.d10	qaddb_36.d10	qaddb_42.d1	qaddb_7.d5	
qaddb_16.d2	qaddb_25.d17	qaddb_28.d1	qaddb_35.d11	qaddb_36.d11	qaddb_42.d2	qaddb_7.d6	



# Obsolete Practice

- Maintaining DB extents less than 2GB:

qaddb.d1	qaddb_16.d3	qaddb_25.d18	qaddb_28.d2	qaddb_35.d12	qaddb_36.d12	qaddb_7.d1	qaddb_7.d7
qaddb.d2	qaddb_17.d1	qaddb_25.d19	qaddb_28.d3	qaddb_35.d13	qaddb_36.d13	qaddb_7.d10	qaddb_7.d8
qaddb.d3	qaddb_17.d2	qaddb_25.d2	qaddb_28.d4	qaddb_35.d14	qaddb_36.d14	qaddb_7.d11	qaddb_7.d9
qaddb.db	qaddb_18.d1	qaddb_25.d3	qaddb_28.d5	qaddb_35.d15	qaddb_36.d15	qaddb_7.d12	qaddb_8.d1
qaddb_10.d1	qaddb_18.d2	qaddb_25.d4	qaddb_28.d6	qaddb_35.d16	qaddb_36.d2	qaddb_7.d13	qaddb_8.d10
qaddb_10.d2	qaddb_19.d1	qaddb_25.d5	qaddb_28.d7	qaddb_35.d17	qaddb_36.d3	qaddb_7.d14	qaddb_8.d11
qaddb_10.d3	qaddb_19.d2	qaddb_25.d6	qaddb_29.d1	qaddb_35.d18	qaddb_36.d4	qaddb_7.d15	qaddb_8.d12
qaddb_11.d1	qaddb_19.d3	qaddb_25.d7	qaddb_29.d2	qaddb_35.d19	qaddb_36.d5	qaddb_7.d16	qaddb_8.d13
qaddb_11.d2	qaddb_20.d1	qaddb_25.d8	qaddb_29.d3	qaddb_35.d2	qaddb_36.d6	qaddb_7.d17	qaddb_8.d14
qaddb_11.d3	qaddb_20.d2	qaddb_25.d9	qaddb_30.d1	qaddb_35.d20	qaddb_36.d7	qaddb_7.d18	qaddb_8.d15
qaddb_11.d4	qaddb_20.d3	qaddb_26.d1	qaddb_30.d2	qaddb_35.d21	qaddb_36.d8	qaddb_7.d19	qaddb_8.d2
qaddb_11.d5	qaddb_21.d1	qaddb_26.d2	qaddb_30.d3	qaddb_35.d22	qaddb_36.d9	qaddb_7.d2	qaddb_8.d3
qaddb_12.d1	qaddb_21.d2	qaddb_26.d3	qaddb_31.d1	qaddb_35.d23	qaddb_37.d1	qaddb_7.d20	qaddb_8.d4
qaddb_12.d2	qaddb_22.d1	qaddb_26.d4	qaddb_31.d2	qaddb_35.d24	qaddb_37.d2	qaddb_7.d21	qaddb_8.d5
qaddb_12.d3	qaddb_22.d2	qaddb_26.d5	qaddb_31.d3	qaddb_35.d25	qaddb_38.d1	qaddb_7.d22	qaddb_8.d6
qaddb_13.d1	qaddb_23.d1	qaddb_26.d6	qaddb_32.d1	qaddb_35.d26	qaddb_38.d2	qaddb_7.d23	qaddb_8.d7
qaddb_13.d2	qaddb_23.d2	qaddb_26.d7	qaddb_32.d2	qaddb_35.d27	qaddb_39.d1	qaddb_7.d24	qaddb_8.d8
qaddb_13.d3	qaddb_24.d1	qaddb_27.d1	qaddb_32.d3	qaddb_35.d28	qaddb_39.d2	qaddb_7.d25	qaddb_8.d9
qaddb_14.d1	qaddb_24.d2	qaddb_27.d10	qaddb_32.d4	qaddb_35.d3	qaddb_39.d3	qaddb_7.d26	qaddb_9.d1
qaddb_14.d2	qaddb_25.d1	qaddb_27.d2	qaddb_33.d1	qaddb_35.d4	qaddb_39.d4	qaddb_7.d27	qaddb_9.d2
qaddb_15.d1	qaddb_25.d10	qaddb_27.d3	qaddb_33.d2	qaddb_35.d5	qaddb_39.d5	qaddb_7.d28	qaddb_9.d3
qaddb_15.d2	qaddb_25.d11	qaddb_27.d4	qaddb_33.d3	qaddb_35.d6	qaddb_40.d1	qaddb_7.d29	qaddb_9.d4
qaddb_15.d3	qaddb_25.d12	qaddb_27.d5	qaddb_34.d1	qaddb_35.d7	qaddb_40.d2	qaddb_7.d30	qaddb_9.d5
qaddb_15.d4	qaddb_25.d13	qaddb_27.d6	qaddb_34.d2	qaddb_35.d8	qaddb_40.d3	qaddb_7.d31	qaddb_9.d6
qaddb_15.d5	qaddb_25.d14	qaddb_27.d7	qaddb_34.d3	qaddb_35.d9	qaddb_41.d1	qaddb_7.d32	qaddb_9.d7
qaddb_15.d6	qaddb_25.d15	qaddb_27.d8	qaddb_35.d1	qaddb_36.d1	qaddb_41.d2	qaddb_7.d33	qaddb_9.d8
qaddb_16.d1	qaddb_25.d16	qaddb_27.d9	qaddb_35.d10	qaddb_36.d10	qaddb_42.d1	qaddb_7.d34	qaddb_9.d9
qaddb_16.d2	qaddb_25.d17	qaddb_28.d1	qaddb_35.d11	qaddb_36.d11	qaddb_42.d2	qaddb_7.d35	qaddb_9.d10



# Why Fixed Extents <2GB?



- There was once a time when hard drives were very large but not so large in capacity:
- We did not have the “Large Files” option
- Older UNIX version limitations: Triple Inode Indirection
- Prevent extents from filling up file system
- Crude way of making sure we stay in fixed extents





# 2GB Fixed Extents

---

- Requires more file handles/descriptors
- Difficult and longer to manage
- The database will crash (or stall with –bistall) once any file reaches 2 GB





# Enable Large Files

---

- To Enable:
  - proutil <DB> -C EnableLargeFiles*
    - While the DB is down
- Available since Progress 9.1C
- Large files must also be enabled at the file system level
- Increase “ulimit” in UNIX

# Agenda

---



- Introduction
- Improve Performance
- Save Time
- Reduce Downtime
- Other Interesting Tricks
- Question



# Obsolete Practice

---

- The following operations can be done online, with some limitations:
  - Modify the schema
  - Add/Modify tables, indexes, sequences, etc...
  - Add DB extents and Storage Areas
  - Modify DB startup parameters:
    - Blocks in database buffers (-B)
    - After-image buffers (-aibufs)
    - Before-image buffers (-bibufs)
    - Lock table entries (-L)
    - Shared-memory overflow size (-Mxs)

# Agenda

---



- Introduction
- Improve Performance
- Save Time
- Reduce Downtime
- Other Interesting Tricks
- Questions





# Other Obsolete Practices

---

- Truncating the DB log: <DB>.lg
  - Traditional Method:  
*cat /dev/null > db.lg*
  - New Method:  
*prolog <DB> -online*
- Avoid protocols that transmit data in clear text:
  - Telnet, FTP, etc...
  - Passwords and data can be “sniffed” with free tools and utilities that may already be installed on the system
  - Use protocols that encrypt the transmission, such as SSH, SFTP and SCP.

# Conclusion



- Who knows how the future will look like?



White Star Software

# Conclusion



- 
- What is useful and has been for years, is not necessarily the best practice today
  - It is worth reviewing to improve performance, save time and avoid downtime by adopting newer methods
  - OpenEdge 10+ (& 11+) lets us manage our DB more efficiently.

# Questions?



# Thank You!

# Try ProTop Free



---

The #1 FREE OpenEdge Database Monitoring Tool



# protop

<http://wss.com/protop.html>

*White Star Software*

# White Star Software



- 
- The oldest and most respected independent DBA consulting firm in the world
  - Four of the world's top OpenEdge DBAs
  - Author of ProTop, the #1 FREE OpenEdge Database Monitoring Tool
    - <http://wss.com/protop.html>