

CA ADS™ Application Performance

Gary Cherlet
IUA/EIUA

IUA/CA IDMS™ Technical Conference
May 16-20, 2016



Abstract

- The CA ADS family of tools is a powerful and fast way to create runtime applications. Efficiency and performance are important considerations. This class will discuss factors that affect runtime performance.

Biography

- Retired after 42 years in computing
- 30 years specializing in CA-IDMS— pre/post sales support for Cullinet (four and a half years) and Allen Systems Group (one year in Singapore)
- Roles— DBA, application developer, security specialist, performance and tuning specialist, IDD customisation and reporting, IDMS/SQL functions and procedures
- Training in all aspects of using IDMS – DBA's and developers – wrote courseware for developers
- Where— Canada, Singapore, England, Malaysia, Thailand, Philippines and Australia

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Size Matters

- Applications
- Dialogs
- Maps
- Subschemas
- Smaller is faster
 - Everything is “table driven”
 - Tables need to be searched
 - Searches consume CPU
- Too small is inefficient also
- As with many things, a balance is best

5

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



How to Reduce Size

- Limit scope of work
- Reduce redundancy
- Eliminate unnecessary pieces
 - Global records (too many elements / too many)
 - Global subschemas
 - Overly large components
 - Dialog code that can be handled automatically (more later)
- No large-scale entities, multiple small ones

6

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Map Design - #1 Way to Manage Size

- Implement standards to control size
 - Header and message formats
 - Control key definitions, responses
- Low number of fields per map
 - Reduce data stream length, faster to build
 - Possibly more interrupts, multiple maps
 - Ease of comprehension
- Use Automatic Editing (whenever possible)
- Use Pageable Maps (when appropriate)
 - Retrieval only
 - Build one page at a time - reduce scratch usage

7

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

8

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Application: Flow of Control

- Define ADSA functions to drive control
 - Eliminate redundant code
 - Provide dynamically-built menus
 - Minimize link levels
 - Prototyping tool
 - Easier maintenance
- Use function keys or 'Execute Next Function'
- Use OPT00086 for security if using EXECUTE NEXT FUNCTION

9

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Applications: Modularize

- Avoid very large applications
 - Define applications in manageable pieces
 - Build 'subset' applications if necessary
 - Use functional divisions
- Build a new control application ("umbrella")
 - Run from primary dictionary or CDMSLIB
 - Next task code in global record or scratch
 - SET new dictname
 - LEAVE ADS NEXT TASK

10

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Application Thread

- List of operative dialogs at run-time
- Operative dialog list
 - Included: when processing control received
 - Removed: when control passes to higher level dialog (including copy of itself)
 - All removed: when control passes out of application
- Record buffers, VDBs, currency blocks held for all operative dialogs (one per level)
- Minimize levels
 - Reduce held storage
 - Simplify debugging & maintenance

Applications: Manage Levels

- First function or dialog establishes top level
- Some commands establish lower levels
 - Invoke
 - Link
 - Ensure there is **always** a **RETURN**
- Multiple link levels
 - Require additional concurrent storage
 - Can be very difficult to debug
 - Can cause waits and deadlocks
- Maximum link level
 - SYSGEN default is ten
 - Attempt to exceed causesabend

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Extended Run-Units

- Occurs with Link, Invoke, Transfer to:
 - Dialog (premap) with no subschema
 - Dialog (premap) with identical schema & subschema
 - User program (Link)
- RATs must be same areas, usage
 - Equal or less restrictive usage
 - Usage Mode, qualifier
- Pro: Less storage, no subschema duplication
- Con: Need to finish and re-bind original dialog
- Use ADS TRACE to understand whether or not new Run Units are being bound – and see the Currency Save/restore operations being performed

Extended Run-Units – visible with ADS Trace and OLP

```

FROM      ON      TO      ON      BIND Run Unit at Top Level
17:01:00  2001-04-11 17:01:22 2001-04-11

. HRID00D. ***** DIALOG-ENTRY ***** **MAIN** 00000000 ENTRY...
. HRID00D. HRID00-PM ***** **MAIN** 00000000 ENTRY...
. HRID00D. HRID00-PM ***** **MAIN** 00002100 OBTAIN . 000790
0000FF08  STATUS=0000 BIND SUBSCHEMA-->EMPSS01 DBN-->EMPDEMO
0000FF08  STATUS=0000 READY AREA RETRIEVAL . AREA-->EMP-DEMO-REGION
0000FF08  STATUS=0000 READY AREA RETRIEVAL . AREA-->EMP-DEMO-REGION
0000FF08  STATUS=0000 BIND REC-->DEPARTMENT
0000FF08  STATUS=0000 BIND REC-->EMPLOYEE
03153E01  STATUS=0000 OBTAIN CALC . REC-->EMPLOYEE
. HRID00D. HRID00-PM ***** **MAIN** 00002700 LINK Command
030D9C01  STATUS=0000 OBTAIN FIRST RECORD IN SET. REC-->EMPLOYEE SET-->
. HRID00D. HRID00-PM . GETJOB . 00000000 ENTRY...
. HRID00D. HRID00-PM . GETJOB . 00005900 LINK . 000BA0
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
. HRID101D ***** DI ***** **MAIN** 00000000 ENTRY...
. HRID101D HRID101-PM ***** **MAIN** 00000000 ENTRY...
. HRID101D HRID101-PM ***** **MAIN** 00000400 OBTAIN . 000394
030D9C01  STATUS=0000 BIND RECORD . REC-->EMPOSITION
030D9C01  STATUS=0000 BIND RECORD . REC-->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000

```

Extended Run-Units – visible with ADS Trace and OLP

```

FROM      ON      TO      ON      COL PRT SKIP  LOG TYPES  ROLL STATUS
17:01:00  2001-04-11 17:01:22 2001-04-11 032 OFF 0021 (WT/TR/DU/ ) 031

030D9C02  STATUS=0000 OBTAIN FIRST RECORD IN SET. R RETURN SET-
. HRID00D. HRID00-PM ***** **MAIN** 00001000 OBTAIN . 00040C
03151D07  STATUS=0000 OBTAIN OWNER OF SET . SET-->JOB-EMPOSITION
. HRID00D. HRID00-PM ***** **MAIN** 00001200 RETURN . 00041C
. HRID00D. HRID00-PM ***** **MAIN** 00004000 ACCEPT . 0009C0
03151D07  STATUS=0000 BIND RECORD . REC-->DEPARTMENT
03151D07  STATUS=0000 BIND RECORD . REC-->EMPLOYEE
CIES RESTORED AND RELEASED FOR DIALOG HRID00D AT LINK LEVEL 00000000
03151D07  STATUS=0000 ACCEPT CURRENT OF RECORD . REC-->EMPLOYEE
. HRID00D. HRID00-PM ***** **MAIN** 00004200 LINK Command
030D7A01  STATUS=0000 OBTAIN NEXT IN SET . REC-->EMPLOYEE SET-->
. HRID00D. HRID00-PM . GETJOB . 00000000 ENTRY...
. HRID00D. HRID00-PM . GETJOB . 00005900 LINK . 000BA0
CIES SAVED FOR DIALOG HRID00D AT LINK LEVEL 00000000
. HRID101D ***** DIALOG-ENTRY ***** **MAIN** 00000000 ENTRY...
. HRID101D HRID101-PM ***** **MAIN** 00000000 ENTRY...
. HRID101D HRID101-PM ***** **MAIN** 00000400 OBTAIN . 000394
030D7A01  STATUS=0000 BIND RECORD . REC-->EMPOSITION
030D7A01  STATUS=0000 BIND RECORD . REC-->JOB
CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000
030D7A02  STATUS=0000 OBTAIN FIRST RECORD IN SET. REC-->EMPOSITION SET-

```


FROM ON TO STATUS

17:01:00 2001-04-11 17:01:22 2001-04-11

—

. HRID101D HRID101-PM 00040C

03151D07 STATUS=0000 OBTAIN OWN SET SET-->JOB-EMPOSITION

. HRID101D HRID101-PM . **MAIN** 00001200 RETURN . 00041C

. HRID00D. HRID00-PM . **MAIN** 00004000 ACCEPT . 0009C0

03151D07 STATUS=0000 BIND

03151D07 STATUS=0000 It looked like when the RU was

CIES RESTORED AND RE from higher level to lower

03151D07 STATUS BIND RU -

. HRID00D. 0000A18

030D8301 IN SET . REC-->EMPLOYEE SET-->

. HRID101D HRID101-PM 0000BA0

. HRID101D HRID101-PM 0000BA0

CIES SA 0000D00D A

. HRID101D HRID101-PM DIALOG-EN

. HRID101D HRID101-PM 000394

. HRID101D HRID101-PM 000394

030D8301 STATUS BIND

030D8301 STATUS=0000 BIND RECORD . REC-->JOB

CIES RESTORED AND KEPT FOR DIALOG HRID00D AT LINK LEVEL 00000000

030D8304 STATUS=0000 OBTAIN FIRST RECORD IN SET. REC-->EMPOSITION SET

. HRID101D HRID101-PM . **MAIN** 00001000 OBTAIN . 00040C

—

https://communities.ca.com/docs/DO C-231151379 - Debugging ADS Applications December 2014 IUA Workshop

- And again and again for each iterative LINK to the lower level and RETURN back to the higher level – saving of many Run Unit BINDs

Restrictiveness	Usage mode	Qualifier
Most	Update	Exclusive
		Protected
	Retrieval	Shared
	Transient Retrieval	
Least	Noready	

Ready Area Table (RAT)

- One for each dialog process in application thread
- Less storage needed for tailored subschemas
- 'Ready All' flag: Ready All, Noready
- Mode: Explicit, ss default, shared retrieval default
- Flag shows ready mode (last command for area)
- READY is not an executable DML command in ADS

Retrieval	X'25'
Update	X'24'
Protected Retrieval	
Exclusive Retrieval	
Protected Update	X'26'
Exclusive Update	X'29'

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Subschema Size

Affects storage for:

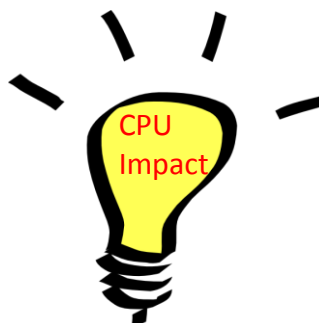
- IB50, VB50 (Subschema control blocks)
- FDB (\$\$SAN Table)
- RAT (Ready Area table) – each occurrence
- **\$*\$CURCY (Currency block)**

Global Subschemas	Tailored Subschemas
Easier to maintain	More maintenance
Demand more space	Reduced storage needs
Ready all areas	Ready fewer areas

Best: semi-tailored application-specific

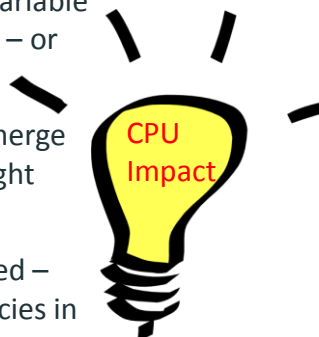
Why large subschemas are a liability

- Currency save/restore processing
- To re-establish run units after a pseudo-converse
- As the application thread lengthens and shortens
- Very CPU intensive



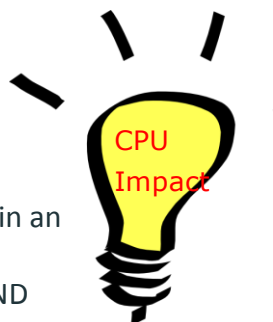
Why so CPU intensive?

- ADS just can't simply copy a subschema variable table (VIB) on a save over a \$CURCY table – or the other way for a restore
- The process is actually more of a logical merge from the top level down – not just a straight copy of the currency tables
- Once the currencies have been rationalized – ADS reestablishes the appropriate currencies in IDMS



Why so CPU intensive?

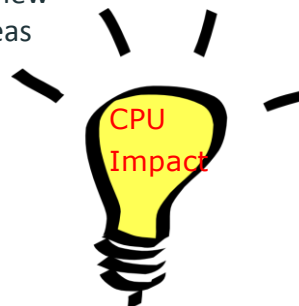
- As currencies are reestablished appropriate locks are reacquired
- This explains why the record being retrieved in an ADS dialog is sometimes different than the “error record” that may have caused an ABEND
- Even the ADS manuals admit that this can take “thousands” of instructions. Basically, be aware that:
 - Cost goes up with an increase in the number of levels
 - Cost goes up with increasing size of subschema(s)



A True Story

Heavy use of Logical Record Facility (LRF)
As LR's needed additional components, or as new LR's were required, additional records and areas were simply added to the subschema.

- The end result of this was:
 - Very large subschemas
 - **An Application that was VERY expensive to run**
- What was the solution?
 - Tailor the subschemas
 - Match needs of logical groups of programs
- What was the result?
 - **A 25% cost reduction for the application**
 - Without changing a single line of ADS code!



Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Dialog Standards

- Symbol table on in test, not in production
- Diagnostic tables always unless very stable
- Limit dialog size to 50k whenever possible
 - Limit to one function (edit, update, etc.)
- Use ADSA responses
 - Default, AGR-Current-Response

Dialog Standards (cont.)

- Perform efficient one pass editing
- Reduce error handling code (e.g. Modify map)
- Eliminate redundant code (common routines)
- Efficient database access
- Minimize work record storage

Efficient Editing

- Editing operator input: 1/3 of processing
- Map edits eliminate dialog edit code
 - Edit tables validate input
 - Code tables translate values
 - Edit modules format input/output data
 - Automatic editing sets error flags, attributes, etc.

Efficient Editing (cont.)

- Execute On Edit Error
 - Dialog gets control instead of mapout
 - Allows edit of multiple fields
 - Easy for user, code must check IN ERROR
- Dialog editing
 - Supplemental editing
 - Set EDIT IS ERROR (errors caught by Code)
 - More flexible
 - More expensive

Eliminate Redundant Code

- Use “Enter” or “Default” Response
 - Define in ADSA
 - Use to handle all editing, whatever key hit
 - Removes redundant checks in responses
- Subroutines
 - If code used more than twice in process
- Include Modules
 - Standard routines
 - Often Subroutines used in multiple dialogs
- Mapless dialogs
 - To “Include” or to “Link” –
that is the Question

Eliminate Redundant Code (cont.)

- CA-supplied or user-written BIFs
 - Dates (Today, tomorrow, CGDATE, etc.)
 - “Date”X BIFs for Y2K dates
 - Arithmetic (Abs value, invert sign, trig, etc.)
 - String (Concat, Substr, Replace, etc.)
- **Avoid heavy nesting**
 - Use User Written BIFs if needed
 - Eliminate redundant or complex code
 - Avoid nesting by building multiple functions into one

<https://communities.ca.com/community/ca-idms-ia-eiua/blog/2014/12/04/if-you-are-planning-on-attending-creative-application-tuning-techniques-at-iaa-workshop-read-this> - for discussions on use of INCLUDE vs LINK and use of User Built-in-Functions

WORK Record Design

- Tailored for each dialog
 - Avoids wasted buffer space
 - Allows for easy dialog modification
 - Saves recompiling entire application
- Include FILLER only if the dialog is dynamic
- New dialogs should have one work record
 - Remember the requirements of “Included code”
- Work records may be shared – eg. common routines – Subroutines in INCLUDED code
- Global records
 - May require Filler to avoid recompiles
 - Can be quite wasteful

33 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



WORK Record Design (cont.)

- Always use COMP or COMP-3 instead of DISPLAY for numeric items which are used for arithmetic operations
- Always use COMP for numeric items that are used as subscripts
- Always place signs on numeric fields that are used for arithmetic (i.e. S9(n) COMP or COMP-3 instead of 9(n) COMP or COMP-3)
- When to modify existing records?
 - Overhead for new record may be greater than the new elements that are required for the dialog

34 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Extended Run-Unit Issues

- Database currencies passed to lower level dialog, restored on return
- Currencies not passed to user programs
- Record locks in upper-level dialog not released
- If lower-level program opens transaction, deadlock possible
- Avoid deadlock by
 - Issue Commit prior to Link (can't rollback)
 - Pass Subschema-ctrl to user program, issue no Bind or Finish in user program
- Reduces CPU by avoiding FINISH then BIND-FINISH on LINK and possibly new BIND on RETURN

LINK NOSAVE

- NOSAVE is faster than save/restore
- Use with caution, especially on modifications
 - Copy of currency passed from caller
 - Currency passed back up, not restored
 - Retrieval NOLOCK dialog to update dialog
- Currencies from before the link are not saved
- Currencies on return from LINK NOSAVE
 - If run-unit is extended, currencies are 'passed upward'
 - Else, currencies are initialized

Database Access

- Largest impact on response time
- Test access
 - CA-OLQ® Online Query for CA-IDMS®
 - CA-IDMS® Database DML Online Option
 - OCF
- Avoid area sweeps, totals, walking long sets
- Use batch jobs for long or inefficient access

Database Access (cont.)

- Physical database tuning important
 - Maintain indexes
 - Orphans
 - Sizing
 - Monitor statistics
 - Reduce calc/via overflows
 - Buffers
- Avoid OOK records to own high activity sets

SCRATCH Records

- Alternative to repeated database I/O
 - Uses only necessary space
 - Avoids over-allocation when data varies
- Efficient for temporary storage
 - Storage pool use, no nucleus calls (I/O)
 - Can be put into XA storage
- Data transfer between Applications (global records)
- Con - requires coding effort
- Avoid Queue records for application storage

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Locking Considerations

- Locks control simultaneous access
- Prevent overlapping updates, broken sets
- Wait: need lock, a competing lock exists
- Deadlock: shared access, exclusive upgrade
- Locking goals:
 - Minimize deadlocks, waits
 - Maximize data integrity
- Options:
 - Start at design phase
 - Access and update “protocols”
 - Don’t lock
 - Prevent simultaneous access
 - Monitor, manage, repair

Don't Lock: RETRIEVAL NOLOCK

- Intended for dialogs to access static data only
- Possible for dialogs with no database update
- Allowed by SYSTEM statement
- Implemented by dialog
 - Dialog Options screen in ADSC
 - Specify RETRIEVAL LOCK: NO
- Saves storage for locking control
- Improves performance
- Use with caution, may result in ‘dirty reads’

Locking Within the Run-Unit

- Set locks explicitly for the duration of R-U
 - KEEP command or keyword (shared lock)
 - KEEP EXCLUSIVE (exclusive lock)
- Locks set implicitly by ready mode & DML
 - Subschema default for area
 - Default: shared retrieval
- Area locks (NL, IS, IX, S, U, UIX, X, XR)
- Record/Row locks
 - Shared (upon read) enable retrieval, not update
 - Exclusive (upon update) prevents other access
 - Area S (protected retrieval) = S lock on every record
 - Area X (exclusive) = X lock on every row

Locking Across Run-Units / Tasks

- Keep Longterm (Notify) Lock options
- Can be used to single-thread
- Two common uses during a pseudo-converse
 - Gain exclusive control of a frequently updated record
 - Monitor updates to infrequently updated records

Locking Across Run-Units (cont.)

Premap Process:

OBTAIN CALC CUSTOMER.

KEEP LONGTERM LOCK-ID EXCLUSIVE
CUSTOMER.

DISPLAY.

Response Process:

MODIFY CUSTOMER.

KEEP LONGTERM LOCK-ID RELEASE.

DISPLAY MSG TEXT 'Customer Modified'.

See also "Slide 45: Sample Code" in session Handout

Locking Across Run-Units (cont.)

- KEEP LONGTERM <longterm-id> RELEASE
 - Lock already released, yields error 1521
 - Releases all long-term locks linked to lterm
 - If no locks held, still returns 0000 status
- KEEP LONGTERM ALL RELEASE
 - Normal exit from application
 - Begin premap response, to cleanup from abends
- Consider using NOTIFY
 - Create a work field to hold the notification
 - Test against the notification field
 - Code example in attachment

KEEP LONGTERM Considerations

- Lock attached to the logical terminal (LTERM)
- **Not** released by COMMIT
- Release required (often forgotten)
 - When the user signs off
 - DCMT V LTERM <ltermid> RES DELETE.
- Managed by lock ID: 1 to 16 characters
 - A constant in single quotation marks
 - A variable field containing the lock ID
- Must be unique for the record type

Handling Deadlocks

- Avoid by single threading (3 strategies)
 - OBTAIN KEEP EXCLUSIVE
 - LINK TO PROGRAM 'ENQUEUE'
 - LINKed to dialog to control record update
- Avoid by using Notify Locks
 - Test for other update first
- Recover from them with automatic recovery
 - ALLOWING ('0329')
 - FIND CURRENT ALLOWING ('0329') on control commands
 - Notify Operator and/or DBA

Minimize Locking Problems

- Tune databases (page & area size, indexes)
- Use least exclusive usage mode necessary
- Small transactions (recovery unit of work)
- Efficient dialogs (testing)
- No conversational programming
- No locks held across pseudo-converse
- Impact of Queue (note COMMIT **TASK**)
 - WRITE PRINTER for Job Submissions
 - WRITE PRINTER for Reports
 - **NO** Application data

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Sysgen Recommendations (GI76847)

- ADSO statement parms
 - Activity Log is NO (compile time impact)
 - Dialog Statistics OFF (**run time impact**)
 - Fast Mode Threshold is OFF
 - Record Compression is OFF
 - Resources are Fixed
 - Storage Mode is Calculated (**maybe** see next)
 - **Tune primary and secondary RBB pools**
 - Maximum Links (10 – value of lower number?)
 - Newpage Mapout No (no literals)

Sysgen Recommendations (cont'd)

- System statement parms
 - Retrieval NOLOCK
 - Scratch in XA is YES
 - Reentrant pool sizes
- Do NOT enable ADSALIVE in Production environments
 - **Recent User experience:**
 - Saving of 36% - 48% CPU per task with ADS/Alive disabled
 - Disable by removing USGAFIX and USGADEL as Start-up Autotask
 - ADS/Alive is used for ADS dialog debugging
 - Avoid in PRO

Reentrant Program Pools

- Many uses with CA ADS:
 - ADSOMAIN (built-in functions)
 - TAT (Task Application Table)
 - Application Load Module
 - Dialog Load Module
 - Map (Tables, Edit Modules)
 - Subschema, Access Module
- XA Reentrant pool size:
 - Base on “working set” of application – or –
 - Determine total size of all load modules – make the Reentrant Pool that big – let the OS work out the “working set”

Agenda

- Size Matters
- Application Wide Strategies
- Run Unit Management
- Subschema Size
- Dialog Design Strategies
- Locking Strategies
- Additional Considerations

Session Summary

- Database tuning is imperative
- Smaller = faster: map, subschema, dialog, application
- Tailor when possible: subschema, work records
- Share when possible: reduce redundancies
- Automate as a rule: map edits, autostatus, application-level function control
- Consider locking during design and testing
- No strict coding rules, maximize trade-offs
- Consider efficiency during design and coding
 - The CA-IDMS/DC environment
 - The business in which an application runs

On the IUA/EIUA Community

<https://communities.ca.com/community/ca-idms-iua-eiua/blog/2014/12/04/if-you-are-planning-on-attending-creative-application-tuning-techniques-at-iua-workshop-read-this>

ADS Run-time Performance – General Guidelines 1 of 3

- The following points are made when “performance” is the over-riding design and coding consideration. As discussed earlier, there are other considerations that need to be taken on board by developers – those guidelines are referenced, where appropriate, so that the developer can take into account all of the issues relating to a particular design issue.
 - All LINKs incur overhead – so minimise the number of LINKs wherever possible (see “Cost of A Link”).
 - LINKs involving run-units cost more than LINKs that don’t involve run-units – therefore where possible the mainline dialog should not have a subschema unless it will access a database directly (see “To Pass or Not to Pass Subschema Control”).
 - INCLUDE modules are preferred over LINKed to sub-routines (see “Reuse Strategies for Common Code”).

ADS Run-time Performance – General Guidelines 2 of 3

- DC-Cobol programs are preferred to ADS dialogs (see “Callable Program”).
- Links to non-ADS programs result in the ADS dialog remaining “active” – so the program is not released and re-acquired for the task – thereby gaining some runtime efficiencies. By default a bound run unit for the higher level dialog will remain bound, and if there is no run unit one will not be bound prior to passing control to the Cobol program (in contrast to LINK USING (SUBSCHEMA-CONTROL)) .
- Because of the syntax of the LINK command - when the higher level dialog links to a lower level ADS dialog – the implementation language must always be ADS for the lower level dialog. Thus, If Cobol is used for the subroutine, the implementation language for the subroutine can subsequently be changed to Assembler without changing the higher level dialog.

ADS Run-time Performance – General Guidelines 3 of 3

- Whenever possible it is better to extend run units than not to – however there may be other, overriding concerns.
- Do use the smallest possible subschema on each dialog that retrieves data from a database (see “Subschema Tailoring”).
- Do access data directly whenever possible. If a dialog has a bound run unit it is cheaper to access the required data from within that dialog than it is to LINK to a subroutine to perform the access (see “Access Data Directly”).
- Please see <https://communities.ca.com/community/ca-idms-ia-eiua/blog/2014/12/04/if-you-are-planning-on-attending-creative-application-tuning-techniques-at-iaa-workshop-read-this> to locate the “see this” references in these points!
- These suggestions were provided by IUA/EIUA Community members who have read this paper and provided these suggestions

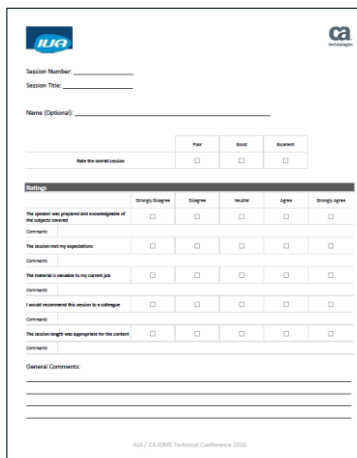
ADS Run-time Performance – Finally

- Run with “Executable Code” – turn off **Symbol Tables**
- Retain **Diagnostic Tables** – aids debugging – minor impact
- Rules for more efficient Cobol code seem to apply equally for ADS code – particularly for Arithmetic processes
- Pay attention to Work record design
- Subschema Size matters
- Number of application “levels” matters
- Dialog Statistics “OFF” unless specifically required for problem dialogs
- **Do NOT enable ADSALIVE in Production environments**

Questions and Answers

Please Complete a Session Evaluation Form

- The number for this session is **A04**
- After completing your session evaluation form, place it in the envelope at the front of the room



The form contains the following sections:

- Session Information:** Session Number, Session Title, Name (Optional).
- Rate the session:** A table with columns for Poor, Good, Excellent and a row for the session rating.
- Rating:** A table with columns for Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree.
- Questions and Answers:** A series of questions with corresponding rating columns.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The speaker was prepared and knowledgeable of the subjects covered	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments:					
The material met my expectations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments:					
The material is valuable to my current job	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments:					
I would recommend this session to a colleague	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments:					
The session length was appropriate for the content	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments:					
- General Comments:** A section for additional feedback.