



Session: J05

Getting Naked with DB2 Attachment (especially in an IMS DB environment)

IDUG 2008
North America

Jack Bailey

BlueCross BlueShield of South Carolina



Experience IDUG

May 20, 2008 • 9:15 a.m. – 10:15 a.m.

Platform: IMS

BlueCross BlueShield of South Carolina is an independent licensee of the Blue Cross and Blue Shield Association.

SQL is the language for communicating application needs to DB2 for z/OS. While most of us deal with the application SQL side or the DB2 subsystem side of this conversation, not many folks discuss how these two actually connect through an attachment method. This presentation discusses the various attachment methods, pros and cons of each, the trials and tribulations of one company's experience with DB2, IMS and MQ series and the how and when you can "Go Commando" with DB2 attachment.

Disclaimer



All information contained in this presentation is based on observations, documentation published by other authors and the authors own individual research.

This information is supplied as is for intellectual discussion only. These are conceptual ideas. Use of this information is completely the responsibility of the user and should be deemed as such.

The information contained in this presentation should be reviewed as a whole. While each item was reviewed for accuracy there are no guarantees.

The important point is that these are concepts.

Objectives



- **Objective 1:** What are the various attachment methods and how are they used?
- **Objective 2:** Why has this become a problem? What is different about today's applications especially when IMS and DB2 are mixed in the equation?
- **Objective 3:** How can you free yourself from imbedded attachment with a generic approach that solves mixed attachment problems?
- **Objective 4:** Do NOT ignore the importance of two-phased or coordinated commits! Why is this a disaster waiting to happen?
- **Objective 5:** When you should NOT free yourself from imbedded attachment and put IMS in charge. When and Why?

What is attachment ?



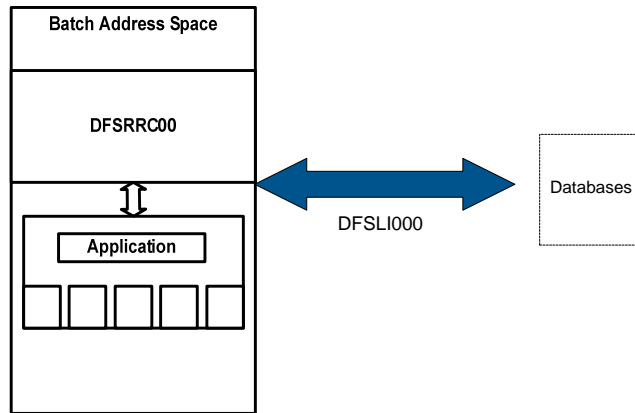
Attachment is simply the creation and maintenance of the communication path between an application and a database management system while the application is active.

What is attachment ? IMS/DB



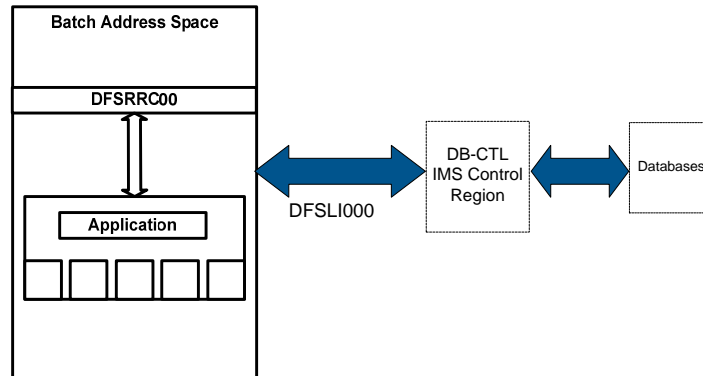
- IMS DB – As IMS is always in control of the application, there is really no such thing as Attachment to an IMS database manager.
- IMS actually attaches to the application in a Batch environment.
- IMS does, however, contain an IMS application interface module known as DFSLI000.

Attachment – IMS/DB DLI



```
//STEPS010 EXEC PGM=DFSRR00,  
// PARM=(DLI,&S010PGM,&S010PSB,.....
```

Attachment – IMS/DB BMP



```
//STEPS010 EXEC PGM=DFSRRC00,  
// PARM=(BMP,&S010PGM,&S010PSB,.....
```

What is attachment ? DB2



- DB2 is a database management system that is totally independent of any application.
- Applications suddenly have to be Attachment Aware.
- Attachment in DB2 is dependent on the Environment.
- Most installations implement the Attachment in such a way as to minimize this awareness to developers.

What is attachment ? DB2



Let us count the ways !

DSNHLI is the standard "High level Interface" name used in most DB2 attachment call statements.

- TSO Attach – DSNELI
- CALL Attach (CAF) – DSNALI
(3rd Party – Softbase's Database Attach)
- IMS Attach – DFSLI000 (IMS language interface module)
- CICS Attach – DSNCLI
- Recoverable Resource Services Attach (RRSAF) – DSNRLI

TSO Attach - DSNELI



- Traditional Vanilla DB2 out of the box
- Batch Job executes IKJEFT01 (TSO Monitor program) to invoke commands
- Commands are passed to the TSO Monitor program through a SYSTSIN DD

```
DSN SYSTEM(DB2O)                - DB2O being DB2 subsystem
RUN PROGRAM(ApplPgm) PLAN(ApplPlan)
END
```

- Commands are serially processed.
 - DSN Command causes the attachment to DB2
 - RUN Command transfers control to an application program
 - Application program must use DSNELI as the Attachment for communication with DB2
 - END completes the step and terminates Connection to DB2
- The TSO monitor program intercepts all exceptions and reports them as condition codes unless alternate programs IKJEFT1A or IJKEFT1B are used.

CALL Attach (CAF) – DSNALI



- Coding your own connects
- Typically third party product is used to reduce application coding – ie. Softbase's Database Attach
- Batch Job executes Application program – No IKJEFT01 (TSO Monitor program) overhead
- Execution is not limited to DB2 availability
- Typically commands (exp 3rd party) passed to through a SYSTSIN DD to maintain consistency of Batch JCL

```
DSN SYSTEM(DB2O)                                - DB2O being DB2 subsystem
RUN PROGRAM(App/Pgm) PLAN(App/Plan)
END
```
- Commands are serially processed.
 - DSN Command causes the attachment to DB2 WHEN first SQL/CONNECT is encountered
 - RUN Command supplies the DB2 Plan for Attachment / Program is ignored
 - Application program must use DSNALI (or 3rd party name) as the Attachment for communication with DB2
 - END has no meaning

IMS Attach – DFSLI000



DFSLI000 is the IMS language Interface module.

DFSLI000 is the IMS Stub for all application interface
not just DB2.

- IMS Checkpoint / Restart facility
- Coordinated commits
- Coordinated recovery
- Same coding as any IMS/DB program
(IMS does all of the work)
- TWO ways to invoke DFSLI000 from BATCH.

IMS Attach – DFSLI000 DL/I



Batch Job executes DFSRRC00 but passes DSNMTV01 (DB2 Interface program) as program to invoke.

- All application programs attempting to access DB2 must be statically linked.
- Execution is limited to DB2 availability
- Typically commands passed to through a DDITV02 DD

DB2O,SYS1,DSNMIN10,,R,-,AMMSJ16D,AMMSA100,AMMSD100

Some parameters are:

- PARM 1 - "DB2O" names DB2 for the attachment to DB2
- PARM 7 - "AMMSJ16D" names the DB2 Thread name for attachment to DB2
- PARM 8 - "AMMSA100" names the DB2 Plan for attachment to DB2
- PARM 9 - "AMMSD100" contains the name of the application program invoked by DSNMTV01

DB2 Version 8 **Application Programming and SQL Guide**

Chapter 23. Processing DL/I batch applications

Page 568

Requirements for using DB2 in a DL/I batch job

Using DB2 in a DL/I batch job requires the following changes to the application program ...:

You must add SQL statements to your application program to gain access to DB2 data.

Is it me ???

IMS Attach – DFSLI000 BMP



Batch Job executes DFSRRC00 but passes Application program name as program to invoke.

- No restriction on static calls.
- Execution is not limited to DB2 availability
- Typically commands passed to through a SYSTSIN DD

```
DSN SYSTEM(DB2O)
RUN PROGRAM(ITCP2016) PLAN(ITCP2016)
END
```

- Commands are serially processed.
 - DSN Command causes the attachment to DB2 WHEN first SQL/CONNECT is encountered
 - RUN Command supplies the DB2 Plan for Attachment / Program is ignored
 - Application program must use DFSLI000 as the Attachment for communication with DB2
 - END has no meaning

CICS Attach – DSNCLI



- Traditional Vanilla CICS DB2 out of the box
- DSNCLI was an attachment method originally shipped with DB2 but ownership of the module changed hands from DB2 to CICS and is now shipped with CICS.
- Coordinated commits / Syncpoints
- Coordinated recovery
- Same coding as any CICS program (i.e. SYNCPOINT, ROLLBACK, etc.)
- Connection is made during startup of CICS region.
- Attachments are by CICS transaction

How do we Decide Attachment ?

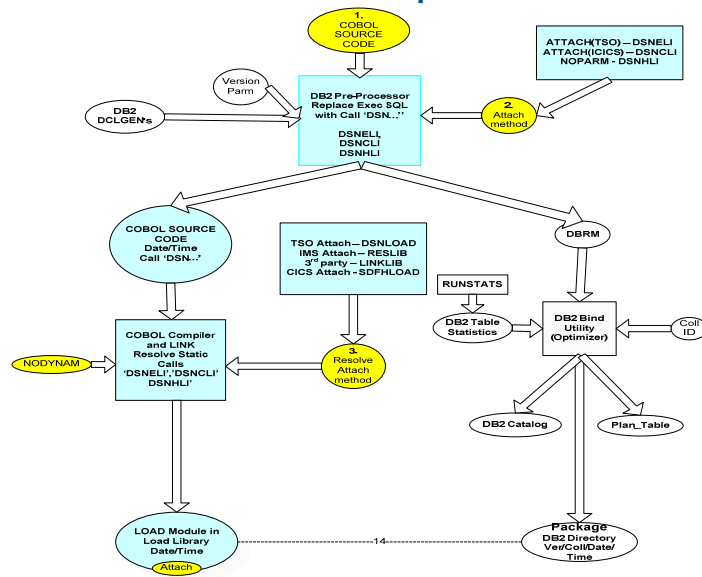


In a typical installations, a technical support area has at one time setup a compile and link process that determines the Attachment.

This would be compile and link JCL supplied stand alone or through a Library Control process.

Developers and programmers have tended in the past to be unconcerned with Attachment.

Traditional DB2 Compile & Link



Rules of the Road



- One and Only One attachment method can be used by a specific instance or execution of an application to establish a connection to DB2.
- For Batch, this means the step within the job can only have one attachment method.
- Once the attachment is established then all attempts to attach with another method will fail.

Sounds Good



So What's the Problem ?

Mixed Attachment Errors



- U0100

A failure occurred during MVS cross-system extended services (XES)
DBA0150E - DB2 CONNECTION FAILURE WITH AN UNRECOGNIZED
REASCODE. REASON CODE = 00F30049

A request was received to connect the requester's TCB to a
DB2 subsystem, but the TCB is already connected.

- SQL -927

The Language Interface (LI) was called when the connecting environment
was not established. The program should be invoked under the DSN
command.

- S04E

When the system detects an error in internal IBM Database 2 (DB2[®])
processing. It will abnormally end a connected user task (TCB) or one of
its own internal tasks. For information about the reason associated with
this abend code, see DB2 Messages and Codes.

So.....

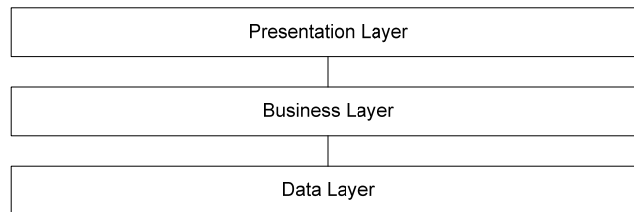


How does that happen ?

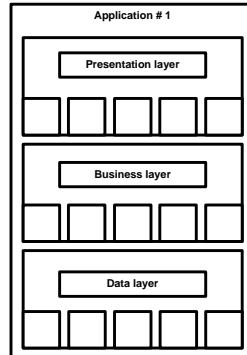
Application Development Strategy



Loose Coupling / Modular Design / Reusability
Separation of presentation, business logic and data access modules that are reusable by various parts of the application or other applications.



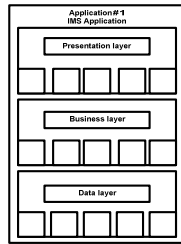
Application Development Strategy



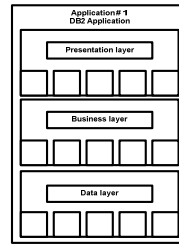
App Development Strategy



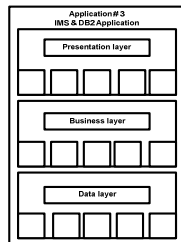
IMS
DFSLI000



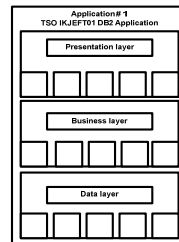
CAF/
DB2
DSNALI



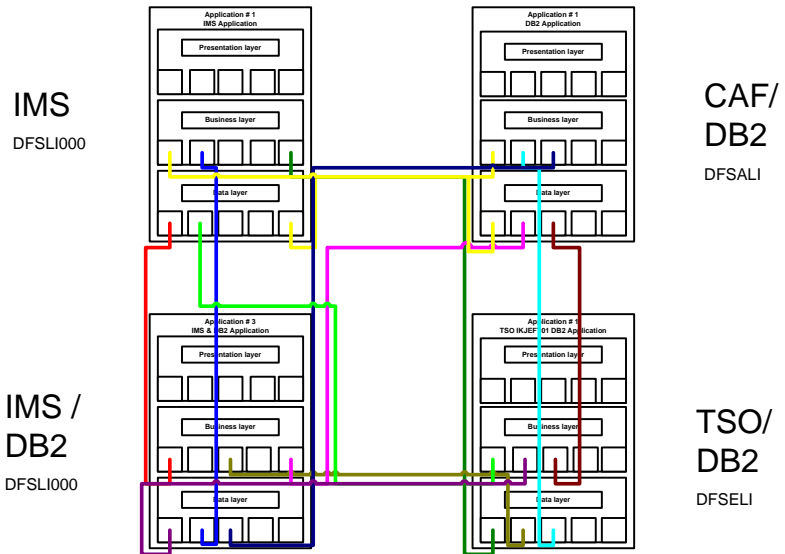
IMS /
DB2
DFSLI000



TSO/
DB2
DSNELI



App Development Strategy



What's the Solution ?

One Solution – Linkage Editor



- INCLUDE DFSLI000
 - REPLACE DSNHLI
 - INCLUDE LOAD(RL55D008)
 - ENTRY RL55D008
 - NAME RL55D008(R)
-
- INCLUDE DSNELI
 - REPLACE DSNELI
 - INCLUDE LOAD(RL55D008)
 - ENTRY RL55D008
 - NAME RL55E008(R)

Problems with approach



- Source and object name mismatches
- Dual objects for a single source
- Remembering to Link alternate copy
- Research into objects with no matching source
- Confusion about what these link decks accomplish

Uggh!

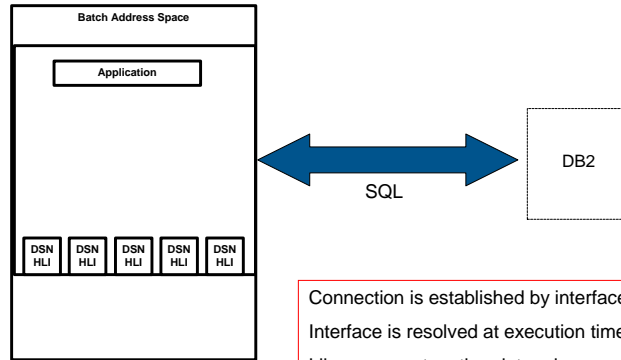


Got another Solution ?

Generic Attach

- Lack of ANY imbedded Attachment method
- Determining your Attachment method at Execution Time
- Originally I called this concept the “No Attach” method until I read an article by Dan Luksetich of YL&A in which he referred to this technique as “Generic Attach”. Obviously a much better definition.

Solution Generic attachment!

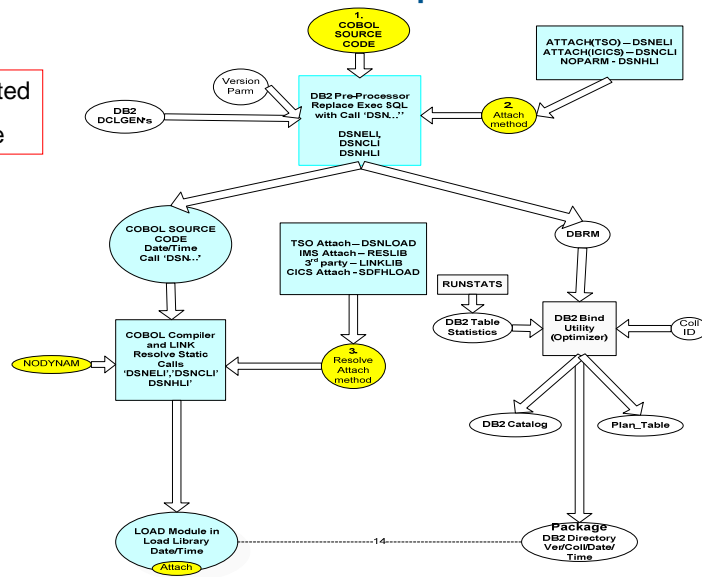


Connection is established by interface type.
Interface is resolved at execution time.
Library concatenation determines attachment method used.

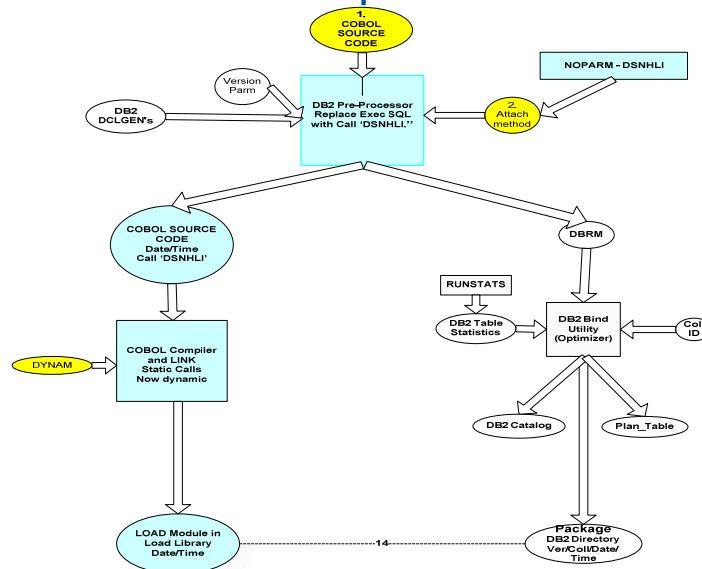
Traditional DB2 Compile & Link



Repeated Slide



Generic DB2 Compile & Link



BUT – How does it work ?



- All Attachment interface calls are to the High Level Interface generic name DSNHLI
- The call to DSNHLI is dynamic.
- The library concatenation controls the Attachment method used.
- DSNHLI should be (if not already) an ALIAS in each of the libraries containing an Attachment method program.

Generic DSNHLI ALIAS of



- DB2 SYS2.DB2.DB20.EXCP.DSNLOAD

Name	Prompt	Alias-of
DSNHLI		DSNELI
DSNELI		

- IMS SYS2.IMSVS.RESLIB

Name	Prompt	Alias-of
DSNHLI		DFSLI000
DFSLI000		

- CAF SYS3.DBATACH.LOADLIB

Name	Prompt	Alias-of
DSNHLI		DBAHLI
DBAHLI		

(3rd Party contains calls to DSNALI)

Advantages of Generic Attach



- Developer of the DB2 module is now totally unconcerned with the attachment method.
- Same source / object code module may now be used regardless of Attachment method.
- Batch – Simple JCL library concatenation controls the Attachment method.
- Developer of the root calling module, who would own the JCL, is now in full control of the Attachment method used (as it should be).

The Naked Truth



Like any thing else in life...

There are times to “Get Naked”

and more importantly

There are times NOT to.

- Understand the difference

Keep your clothes on if...



You need coordinated or two-phased commits
between DB2, IMS and MQ updaters.

- It is as simple as that !

Okay so...



No big deal... Right ?

Let me Repeat
Keep your clothes on if...



You need co-ordinated or two-phased commits between DB2, IMS and MQ updaters.

Who Does that ?



IMS and CICS are two subsystems that are coordinator subsystems.

They will coordinate commits between participant subsystems.

CICS will act as a coordinator to IMS.
(We will just eliminate CICS from further discussion to simplify things. The concepts are the same.)

Power of IMS Attach



- DB2 is oblivious of updates to others (DB2 does not play well with others.)
- IMS, on the other hand, does play well with others.
- IMS will coordinate all of the outstanding commits and issue them across the board (i.e. We all commit or we all fail.)

IMS is "MOM" while DB2 is the mischievous child prodigy.

Coordinated/two-phased commits

If IMS coordinates commits between IMS, DB2 and MQ, wouldn't that be a three phase commit ?

NO - Coordinated or two-phased commits works as follows:

One system acts as a coordinator of all updates.
All other systems are known as participants.

At commit time, there is a two-phase process to committing:

Phase I - The coordinator notifies each system of an intent to commit.
Each system independently determines if enough information has been recorded and whether the outstanding work can be committed.

Phase II – After all systems have reported back in Phase I that all of the outstanding work can be committed, then the coordinator issues the commit to all systems. Should any system abend during phase II, then the commit operation would complete as part of that systems recovery process.

Again...



**IF you need coordinated commits
use IMS Attach !**

What you talking about Willis?



Real Life Story

A batch application was developed that processes an MQ Queue of messages and updates IMS databases and DB2 tables

- Main Application Loop
 - Perform until Stop message on Queue
 - Destructive read of the MQ message
 - Call module to perform IMS Updates
 - Call module to perform DB2 Updates
 - Take an IMS checkpoint

Setting the scene!



- Job ran for 24 hours processing MQ messages
- Hundred's of thousands of messages processed
- All modules are called dynamically.
- Bad data on message caused application program to S0C7.

No problem ! Updates committed! IDUG The Mainframe DB2 User Community

- IMS takes care of all coordinated or two-phased commits during checkpoint processing.
- When a checkpoint is taken IMS commits the destructive read to the MQ Queue, commits the updates to the IMS database and commits the updates to the DB2 tables.
- Cool Huh !

Long Backout! What's Up!



- Did we mention we were using Call Attach (CAF)?
- NOTE: IMS coordinates the commit of all DB2 updates IF and only IF IMS knows about them (i.e. DFSLI000)
- As NONE of the DB2 updates were EVER committed, the job began rolling back 24 hours of DB2 updates and mounting log tapes.
- 12 hours later the job ended

Uh Oh!



- Never say things cannot get worse!
- All of the IMS updates had been applied but none of the DB2 updates and the input (MQ) was gone.
- What do we do now !
- This mistake was only made **once**.

What did we learn from this !



A simple set of Rules

- IMS programs that read DB2 always use imbedded IMS Attach not Generic attach.
- IMS Driver programs that will be calling any possible DB2 modules should be modified to be DB2. This is as simple as issuing an SQL Select Current Date in the initialization section. This will establish the Attachment method and will ensure that no subsequent called DB2 module could use an incorrect Attachment method.

Real world



- Currently we have one entire application in production using this process and we have been extremely successful.
- One other application is undergoing this methodology.
- We have a technical DB2 Attachment committee composed of Database Administration, Database Application Support, Technical Systems Support and Applications to support implementation of this technique as a shop wide standard.

Caveat



- JCL Joblib/Steplib includes
- One discovery we had was that life is much simpler if batch Jobs are broken down so that all of the steps use the same Attachment method. The Joblib or a Joblib include then easily controls the method.
- Unfortunately, if you mix methods in different steps then you would need a steplib or steplib include which is a little more awkward.

Session J05



Getting Naked with DB2
Attachment methods

Jack Bailey

BlueCross BlueShield of South Carolina

jack.bailey@bcssc.com