An IMS View of DBCTL

David Compton

Terminology and Trademarks

⚐ Terminology

- ►RRS - Resource Recovery Services
- ►ODBA - Open DataBase Access
- ►DRA - Database Resource Adapter
- ►AAS - Application Address Space
- ►AIB - Application Interface Block
- ►CCTL - Coordinator Controller

⚐ Trademarks

- ►MVS/ESA
- ►IMS/ESA*
- ►DB2*
- ►S/390*
- ►ESA/390
- ►IBM*
- ►IBM COBOL for MVS
- ►System/390*
- ►CICS
- ►CICS/ESA

* Trademarks followed by an asterisk (*) are registered.

Agenda

⚠ **What is DBCTL?**

⚠ **DBCTL Overview**

⚠ **CICS as a CCTL**

⚠ **DBCTL Subsystem structure**

⚠ **DRA Startup table**

⚠ **Syncpoint**

⚠ **DBCTL UORs**

⚠ **DBCTL Indoubts**

⚠ **/DBR & /STO REGION commands**

⚠ **DBCTL Education**

⚠ **Summary**

What is DBCTL?

⚠ **Hierarchical database subsystem**

⚠ **Provides independent database services to a CCTL**

- Most commonly CICS

⚠ **Coordinated recovery with CCTL**

- CCTL is the controller in syncpoint process

- Single or two phase commit protocol is used

DBCTL Overview

⟁ **Database Control Subsystem**

- Full Function Database support and access

- FastPath DEDB support and access

⟁ **IMS DBCTL service provided by**

- IMS Database Manager (IMS DB)

- IMS DB/DC

⟁ **Multiple CCTL connections to one DBCTL Subsystem**

- CCTLs must reside on same MVS image as DBCTL

DBCTL Overview continued

⚠ **BMP support**
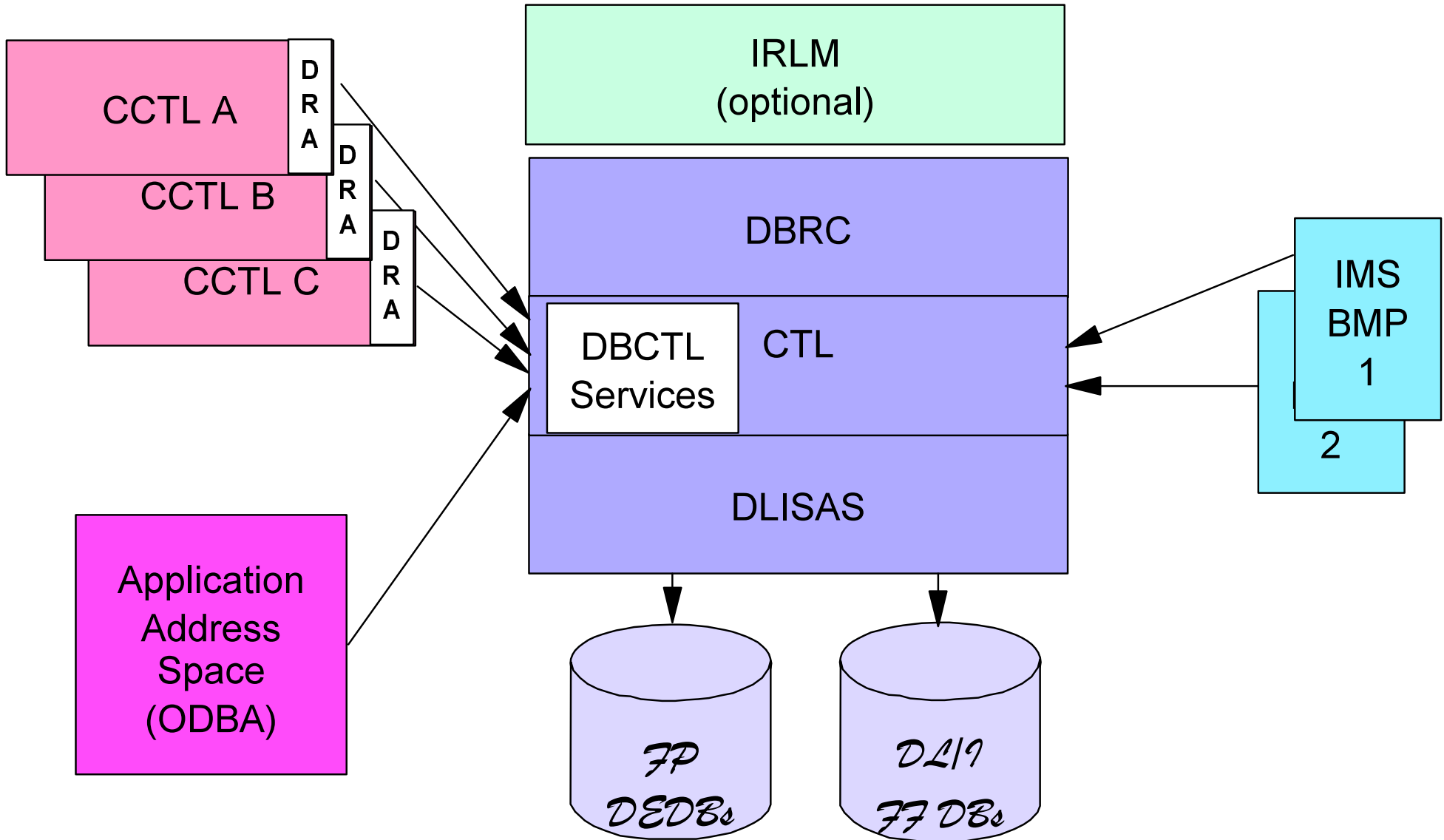
- Non-message driven in a DB only environment

⚠ **Maximum of 999 concurrent applications**

- BMPs, CCTL threads, ODBA threads
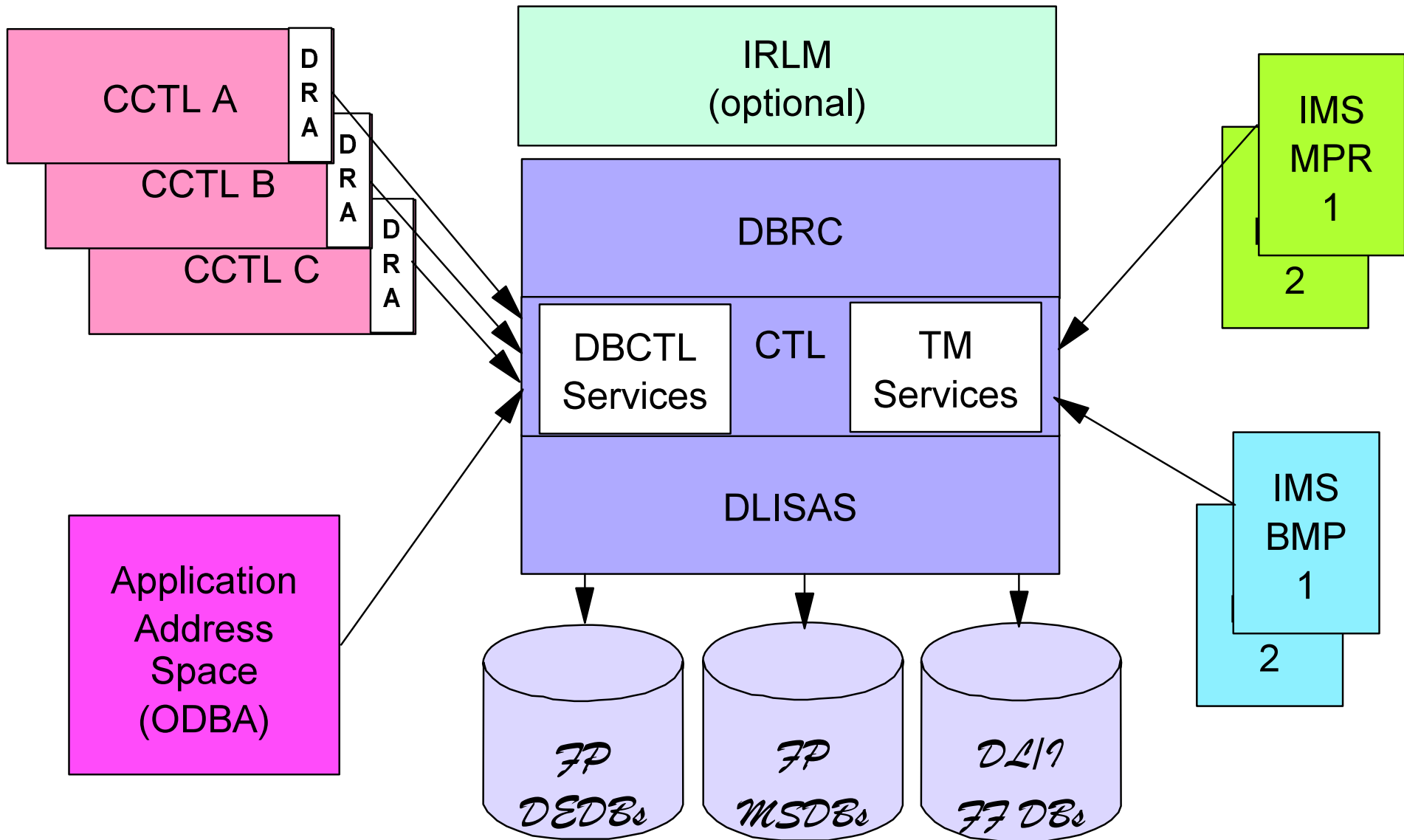
- & MPRs (DB/DC only)

CICS as a CCTL

△ **CICS predominant exploiter of DBCTL services**

△ **CICS must use DBCTL function for IMS DB access**

- IMS V5 Local DL/I support dropped

△ **DBCTL and CICS release independent**

- No CICS "sysgen" required for DBCTL

△ **Multiple CICS subsystems can connect to a single DBCTL**

- CICS can connect to **only** one DBCTL at a time

DBCTL Subsystem Structure

DB/DC Subsystem Structure

CCTL A

CCTL B

CCTL C

D R A

D R A

D R A

Application Address Space (ODBA)

IRLM (optional)

DBRC

DBCTL Services

CTL

TM Services

DLISAS

IMS MPR 1

2

IMS BMP 1

2

FP DEDBs

FP MSDBs

DL/I FF DBs

DRA Startup Table - DFSPZPxx

⚠ **DFSPRP macro defines DRA Startup parms**

⚠ **Assemble DFSPZPxx**

- xx = Any alphanumeric characters

- Default is 00

- DFSPZP00 source is supplied

⚠ **Where can you find the Parm Descriptions**

- DFSPRP DSECT=YES

- Browse DFSPZP00

- IMS/ESA Install Volume 2

DFSPRP parms (trip-up ones anyway)

CNBA

- Total number of FastPath buffers
- FPBUF x MAXTHRD <= CNBA
- Needed for FP DEDB access

MINTHRD & MAXTHRD

- 1 <= MINTHRD => 999
- 1 <= MAXTHRD => 999
- If MAXTHRD < MINTHRD then MAXTHRD = MINTHRD

The Rest of the DFSPRP parms

FUNCLV

▶ Function level - Always 1 (one)

DDNAME

▶ DRA Reslib DDNAME (CCTLDD)

DSNAME

▶ DRA Reslib (IMS.RESLIB)

DBCTLID

▶ IMS/DBCTL IMSID (SYS1)

SOD

▶ Snap Output Dataset (A)

USERID

▶ User Identifier

TIMER

▶ IDENTIFY Timer value (60)

AGN

▶ Application Group Name

TIMEOUT

▶ DRA Termination Timeout Value (60)

IDRETRY

▶ ODBA connection parm

Syncpoint

⚠ **Two Phase Commit**

- ▪ Prepare

- ▪ Commit or Abort

- ▪ Indoubts may result

⚠ **Single Cycle Commit (SCC)**

- ▪ IMS/ESA V6 Enhancement

- ▪ When only ONE Resource Manager's resources have been updated

- ▪ Significant reduction of INDOUBT window

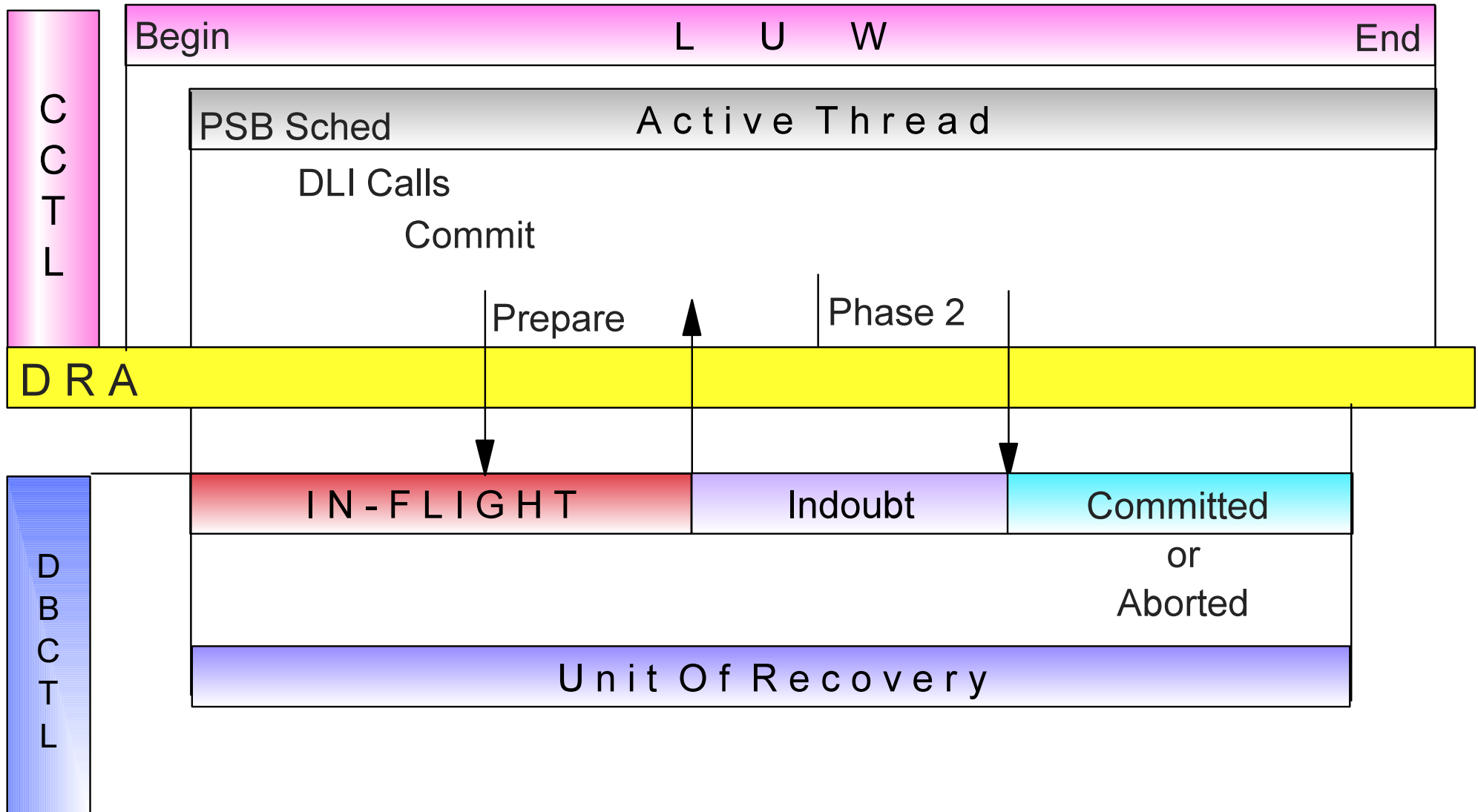- ▪ Improved Syncpoint processing time

Two Phase Commit Overview

⏶ **Coordinator (CCTL/CICS)**

- Coordinates and makes decisions about commit processing (commit or abort)

- Responsible for integrity of its own resources

⏶ **Participant (IMS DBCTL)**

- Takes direction from coordinator (commit or abort)

- Responsible for integrity of its own resources

Two Phase Commit

Begin   L   U   W   End

C C T L

PSB Sched   A c t i v e   T h r e a d

DLI Calls

Commit

Prepare   Phase 2

D R A

I N - F L I G H T   Indoubt   Committed
or
Aborted

D B C T L

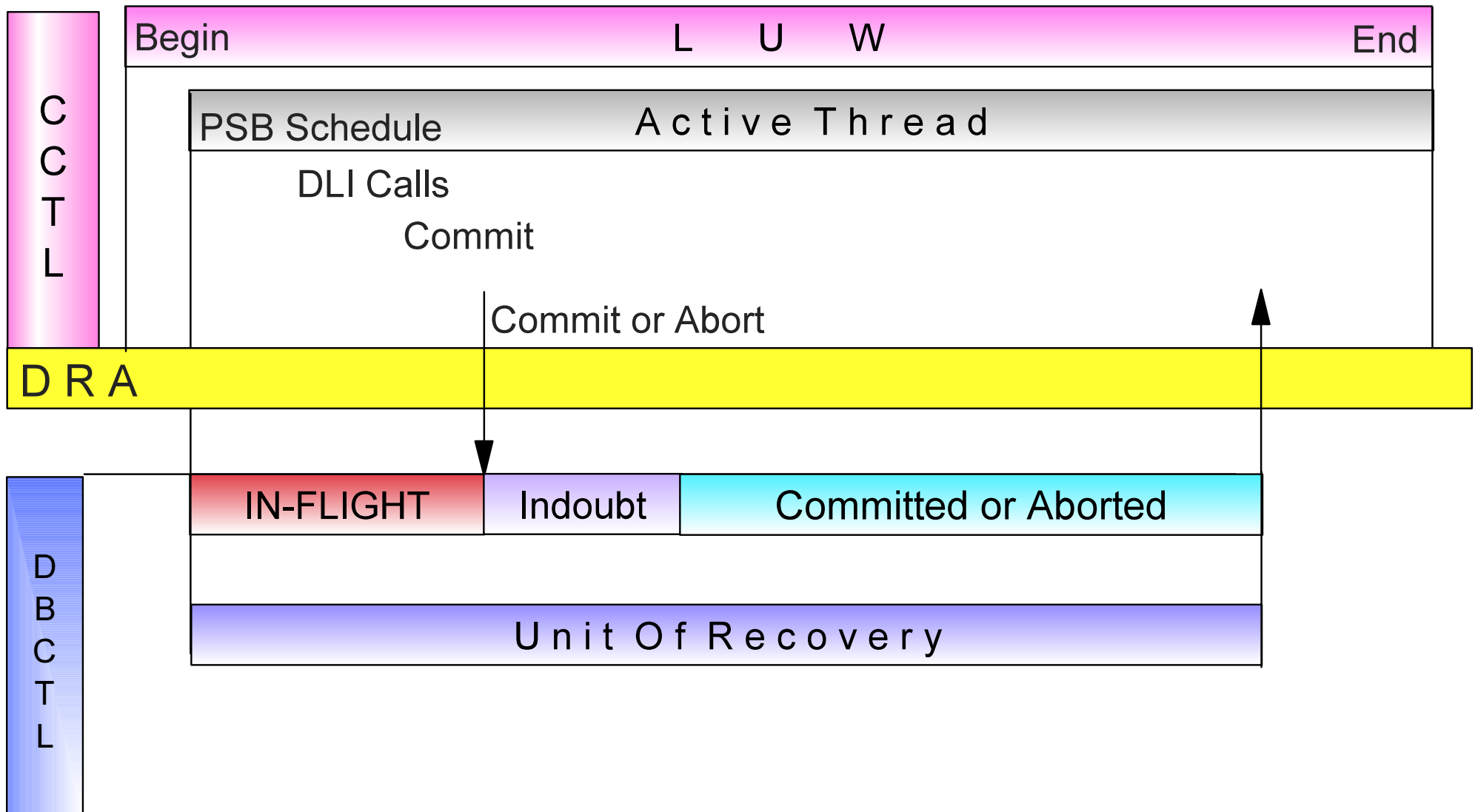U n i t   O f   R e c o v e r y

Single Cycle Commit Overview

⊿ **Coordinator (CICS)**

- Coordinates and makes decisions about commit processing (commit or abort)

- Responsible for integrity of its own resources

- Used when only ONE Resource Manager's resources have been updated.

⊿ **Participant (IMS DBCTL)**

- Takes direction from coordinator (commit or abort)

- Responsible for integrity of its own resources

Single Cycle Commit

C C T L	Begin		L U W	End
	PSB Schedule		A c t i v e T h r e a d	
		DLI Calls		
			Commit	
			Commit or Abort	

D R A

IN-FLIGHT	Indoubt	Committed or Aborted

U n i t   O f   R e c o v e r y

D B C T L

17

Single Cycle Commit

⚠ **Indoubt window only while writing '3730 Start of Phase 2 Commit'**

⚠ **IMS Syncpoint still performs Two Phase Commit**

⚠ **No response to CCTL until both Phases have completed or a failure is detected**

⚠ **PAPLRETC is set with return code of the request**

Commit Processing and Database updates

⚠ **Full Function Updates**

- Full function Database updates written during Phase 1

- In case of abort, backout is performed

⚠ **DEDB Updates**

- Updates performed asynchronously, completed before end of Phase 2

- In case of abort, updates are tossed away, backout is not needed

Phase 1	Indoubt	Phase 2

FFDB

DEDB
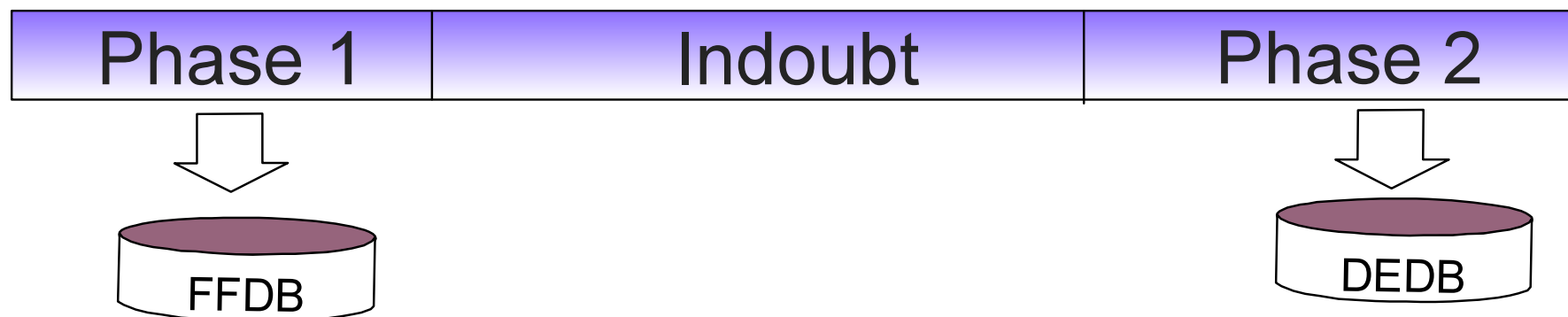
DBCTL UOR

⚜ **UOR = Unit Of Recovery**

⚜ **UOR is represented by unique recovery token**

⚜ **Recovery token is created by CICS after LUW begins**

⚜ **Recovery token is maintained by DBCTL until completion of commit process**

Recovery Token Format

⚠ **DBCTL will not allow identical recovery tokens to be in use at the same time or across consecutive commit cycles**

⚠ **Recovery token is entered in some CICS and DBCTL commands and displayed in some messages**

⚠ **Recovery token format (from CICS):**

- 8 bytes - CICS APPLID

- 8 bytes - Unique UOR ID - Store Clock Value

Status of DBCTL UOR

In-flight

- DBCTL or CICS failed before end of Phase 1
- DBCTL backs out updates dynamically

Indoubt

- CICS or DBCTL failed after end of IMS Phase 1 and before start of IMS Phase 2
- If failed before instant of commit, CICS will tell DBCTL to backout at reconnect
- If failed after instant of commit, CICS will tell DBCTL to commit at reconnect

In-commit

- DBCTL failed after start of IMS Phase 2
- Changes will be committed during restart

In-abort

- DBCTL failed after UOR started backout
- Backout is completed during restart

Recoverable Indoubt Structure (RIS)

⚠ **RIS built for each Indoubt UOR when the CCTL or a CCTL thread fails**

⚠ **RIS built at DBCTL restart if DBCTL fails**

⚠ **RIS used by DBCTL during reconnection to CCTL if Indoubt UORs existed at CCTL/DBCTL failure.**

⚠ **RIS logged at DBCTL system checkpoint**

⚠ **RIS Contents**

- RRE - Recovery token

- IEEQE -Changed data records

- BEEQE - Indicates inaccessible data due to unresolved indoubts

- EQEL - Links RIS elements

Indoubt Resolution

- **Resolution is automatic under most circumstances**
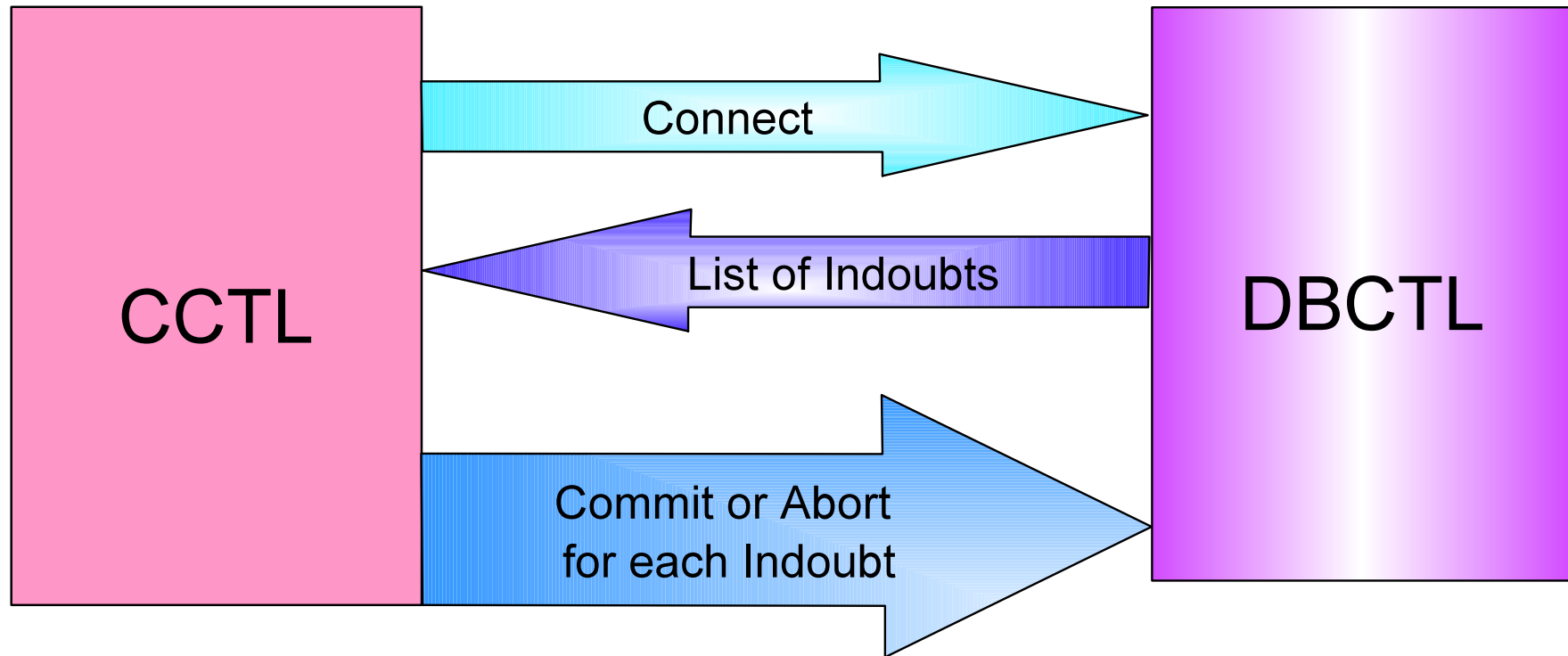 - Done at CCTL reconnection to DBCTL

- **DBCTL command to display indoubts**
  - /DIS CCTL cctlname INDOUBT
  - Displays Pseudo-Recovery token, Recovery token and PSB name

- **DBCTL command to change the status of unresolved indoubts**
  - /CHA CCTL cctlname { prtkn | ALL} { Commit | Abort}
  - prtkn from /DIS CCTL Pseudo-Recovery token
  - Use only if reconnect has not resolved. Verify CCTL's taken action

Automatic Resolution of INDOUBTS

/DISPLAY CCTL INDOUBT

/DIS CCTL CICS1 INDOUBT

CCTL STATUS	PSEUDO-RTKN	RECOVERY-TOKEN	REGID	PSBNAME
CICS1				ATTACHED
	000100C0	9FFA956B7AE24E00	PSBAAA	INDOUBT
	00010040	9FFA9568FF594301	PSBBBB	INDOUBT

⚠ Recovery-Token = Unique token from CICS/CCTL

⚠ Pseudo-Rtkn = DBCTL's token to be used on the /CHANGE command

/CHANGE CCTL

/DIS CCTL CICS1 INDOUBT

CCTL	PSEUDO-RTKN	RECOVERY-TOKEN	REGID	PSBNAME
STATUS				

CICS1
ATTACHED

| | 000100C0 | 9FFA956B7AE24E00 | | PSBAAA | INDOUBT |
| | 00010040 | 9FFA9568FF594301 | | PSBBBB | INDOUBT |

/CHA CCTL 000100C0 ABORT

▶ ABORT updates by PSBAAA

/CHA CCTL 00010040 COMMIT

▶ COMMIT updates by PSBBBB

/DBR, /STOP command Deadlocks

⚠ **/DBR command basics**

- Command rejected for 'long running applications'
  - ▸ BMPs are long running

- Command will wait for 'short running applications'
  - ▸ MPP & CICS/DRA threads are short running

⚠ **Deadlock potential Prior to V5**

- /DBR and /STO region commands under same IMS task

- /DBR targets resources held by CICS/DRA thread

- /STO targets holder of resources

- /STO will not be processed until /DBR completes

- /DBR will not complete until thread is stopped

⚠ **Difficult to determine what thread to stop**

/STOP REGION enhancements V5.1 and up

⚠ **/STOP command processing moves to the SDAB IMS task**

- Eliminates deadlocks with /DBR /DBD commands

- Allows resource contention resolution for /DBR /DBD and threads

⚠ **/STOP ABDUMP sends 'Cancel Lock Request' to IRLM**

- If the lock request is not in/for Backout

- This is not applicable with PI locking

⚠ **IMS APARs required**

- V5 APARS
  - ▸ PQ00893 & PQ17087

- V6 APARS
  - ▸ PQ06544 & PQ17435

⚠ **IRLM 1.5 apar PQ05602**

IMS / IRLM Long lock Detection Report

⧖ **IRLM Longlock Report**

- Requires IRLM 2.1in a SYSPLEX data sharing environment

- RMF Monitor II report ILOCK
  - ‣ 79.15 records
  - ‣ Supported in OS/390 Release 2 and up
  - ‣ Introduced with apar OW28410

⧖ **The report provides information required to allow easy determination of transactions/threads causing the hang**

⧖ **Report contains IMS region | thread number and CICS task id**

- IMS V5 apar - PN84685

- IMS V6 base code

- CICS 4.1 apar - PN84787

- IRLM 2.1 apar - PN79682

IMS Long Lock Detection

⚠ IRLM / IMS considerations

- IRLM APAR PQ15432 for automatic detection of wait for lock

- IMS determines detection cycle at connection to IRLM
  - ▸ Currently set at about 5 minutes
  - ▸ Simple 1 line usermod to change

⚠ RMF requirements

- Must explicitly specify SMF record 79(15) at RMF startup
  - ▸ PARM='MEMBER(00),SMFBUF(SPACE(32M),RECTYPE(70:78,79(15)))'

⚠ Execution

- Message DXR162I issued when long wait detected

- IRLM drives IMS exit

- IMS writes SMF record

IMS Long Lock Detection

⚠ **Use RMF to view data**

- Get to RMF (usually =RMF or RMF from TSO option 6)

- Select option 2 - monitor II

- Select option 3 - resource

- Select option 9 - ILOCK

- Enter ILOCK ALL to view all blockers and waiters

Sample #1 Long Lock Report

```
Cycle    State  Type      IMS_ID  PST_Number  PSB_NAME
         Lock_Name                             CFLock_Structure
         Recovery_Token         Lock_Held_Time
----------------------------------------------------------------
000019BC  B    DBCTL   IMSA      0017          PROGAS1A
         00000126DC800101D700000000000000  IRLMLOCK1
         CICSDAA2ACE7A51B55B16080
CICS_Task_ID
Lock_Count
DB/Area_Name
----------
00000086
00000000
```

Sample #2 Long Lock Report

RMF - ILOCK IRLM Long Lock Detection

```
State    Type       Lock_Name                                    PSB_Name  Elap_Time
CICS_ID
         IMS_ID  Recovery_Token                         PST#   TRX/Job      Wait_Time
DB/Area
---------------------------------------------------------------------
 CF Structure                          at 07/22/1999 15:48:21         Deadlock Cycle
000000A8
---------------------------------------------------------------------
TOP        BMP     0B0000002C800201C600020000000000 FPSBPA      00:00:50
BLOCKER  IM1A    IM1A        0000000100000000      0001    DDLOCK1
---------------------------------------------------------------------
WAITER     BMP     0B0000002C800201C600020000000000 FPSBPA
           IM1A    IM1A        0000000200000000      0002    DDLOCK2     00:00:29
DISTDB
```

IMS DBCTL Education

⬙ **CE63 CICS and DBCTL**

⬙ **CM100 IMS System Programming: DBCTL**

Summary

⚠ **CICS and IMS release independence**

⚠ **Recovery and failure isolation**

⚠ **Syncpoint**

- Normal Two Phase

- Single Cycle Commit support with IMS V6

⚠ **/DBR and /STO THREAD improvements**