

Implications of Threadsafe in CICS Transaction Server

**Ed Addison**

**Session 4090**

## Agenda

- Exploiting the Open Transaction Environment (OTE)
- OPENAPI Support
- Program and Task Related User Exit (TRUE) definitions
- Threadsafe commands
- Steps for ensuring threadsafe
- Checking Concurrency
- Threadsafe Problems

## Exploiting Open Transaction Environment Pre 3.1

- The goal of making programs threadsafe is to enable them to remain on an open TCB. Switching occurs in the following circumstances
  - When a program that is **not** defined as threadsafe makes a DB2 request, CICS switches from the Quasi-Reentrant (QR) TCB to an open TCB, and back to the QR TCB again when the DB2 request is complete.
  - When a user exit program that is **not** defined as threadsafe is used in the course of a DB2 request, CICS switches from the open TCB (where the DB2 request is executing) to the QR TCB. The user exit program is executed on the QR TCB, and then the task is switched back to the open TCB to complete the DB2 request.
    - For example, the XRMIIN and XRMIOU global user exits might be invoked in the course of the DB2 request. If the exit programs are not defined as threadsafe, this TCB switching occurs. If the exit programs are defined as threadsafe, processing will continue throughout on the open TCB.

## Exploiting Open Transaction Environment Pre 3.1

- When a program that is defined as threadsafe is executing on an open TCB and invokes any EXEC CICS commands which are not threadsafe, CICS switches back from the open TCB to the QR TCB to execute the non-threadsafe code.
  - The program then continues to execute on the QR TCB. However, if the program makes any further DB2 requests, CICS must switch back again to the open TCB.
- When a program that is defined as threadsafe and is executing on an open TCB invokes a task-related user exit (TRUE) program which is not defined as threadsafe, CICS switches back to the QR TCB and gives control to the TRUE program.
  - When the task-related user exit program completes processing, the application program continues to execute on the QR TCB, in the same way as it would after issuing a non-threadsafe EXEC CICS command.

## Exploiting Open Transaction Environment Pre 3.1

- When a program that is defined as threadsafe and is executing on an open TCB invokes a threadsafe CICS command, it is possible for a global user exit to be invoked as part of executing the command.
  - CICS switches back to the QR TCB and gives control to the global user exit program.
  - When the user exit program completes processing, CICS switches back to the open TCB to continue processing the threadsafe CICS command.
- When a program that is defined as threadsafe and is executing on an open TCB completes, CICS switches back to the QR TCB for task termination.

## Exploiting Open Transaction Environment Pre 3.1

- To make an application program remain on an open TCB:
  - Ensure that the program's logic is threadsafe. That is, the code between the EXEC CICS commands must be threadsafe. If you define a program to CICS as threadsafe but include application logic that is not threadsafe, the results are unpredictable
  - Ensure that the program uses only threadsafe EXEC CICS commands
  - Ensure that the program is defined to CICS as threadsafe
    - Use the CONCURRENCY attribute of the program resource definition to do this
  - Ensure that any user exit programs in the execution path used by the program are coded to threadsafe standards and defined to CICS as threadsafe
    - This includes dynamic plan exits, global user exits, or task-related user exits
  - If you are coding a user exit program (a global user exit or a task-related user exit), define it as threadsafe so that it can be used on the same L8 TCB as a threadsafe application which calls it



## OPENAPI Support 3.1

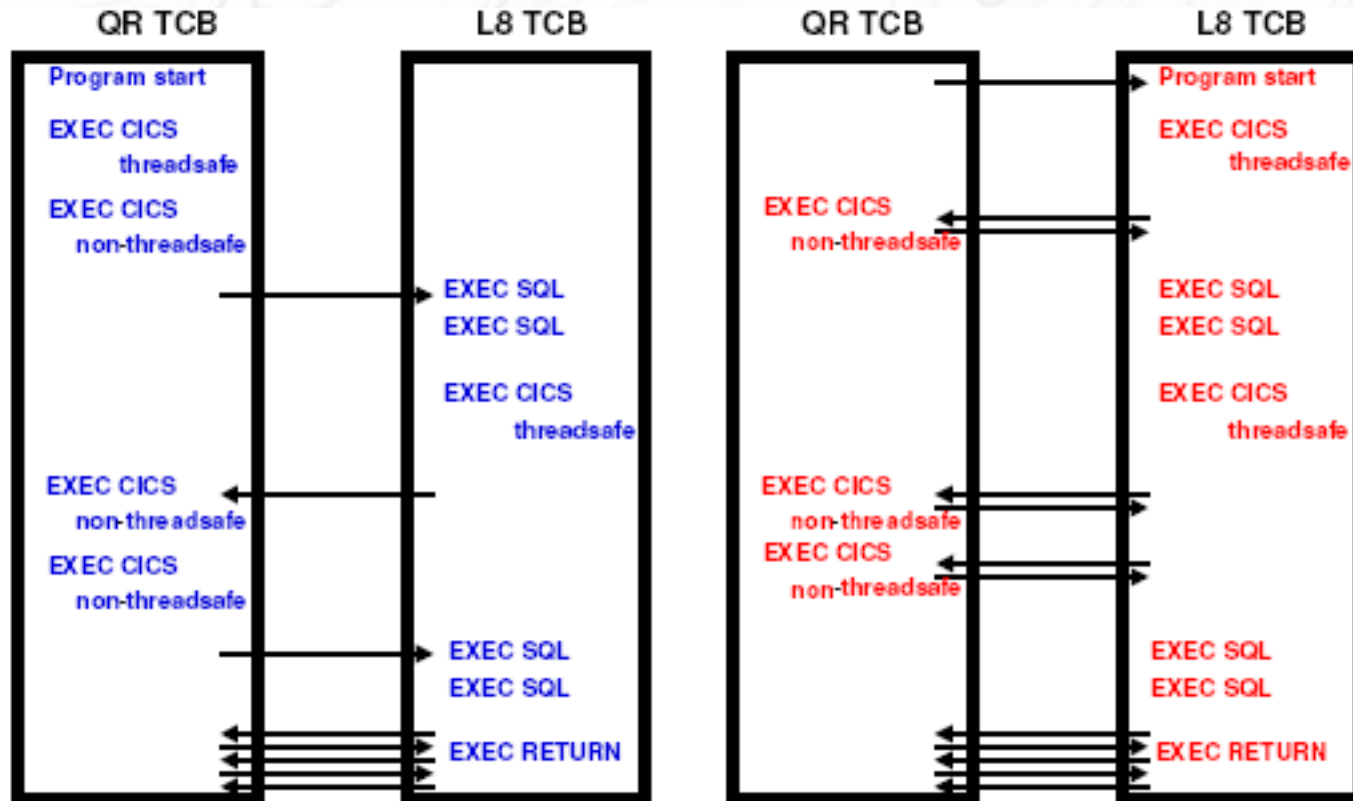
- CICS 3.1 extends the use of Open Transaction Environment (OTE) by providing support for OPENAPI application programs
  - Prior to 3.1 OPENAPI was available only to task related user exits (TRUEs).
- OPENAPI allows an application not only to define itself as threadsafe, but the application must run on an OPEN TCB rather than on the QR TCB.
- Candidate programs for defining as OPENAPI THREADSAFE (assuming their application logic is threadsafe) include:
  - Programs which use CICS threadsafe APIs only (to avoid the double TCB switch) or only limited non threadsafe CICS commands
  - CICS-DB2 applications
  - CPU intensive programs
  - Programs wishing to use other (non CICS) APIs at their own risk

## OPENAPI Support 3.1

- Differences between a CICSAPI QUASIRENT program, a CICSAPI THREADSAFE program and an OPENAPI THREADSAFE program
  - A CICSAPI QUASIRENT program only issues CICS APIs and its application logic is not threadsafe. It always runs on the QR TCB.
  - A CICSAPI THREADSAFE program is capable of running on either the QR TCB or an open TCB because its application logic is threadsafe.
    - This program runs on the QR TCB until some event moves it to an open TCB
      - A call to an OPENAPI TRUE, such as a DB2 call, is an example of an event that would move a CICSAPI THREADSAFE program to an open TCB.
    - After transferring to an open TCB, the program remains there until something forces it back to the QR
      - For example a non threadsafe CICS API call. If this happens the program remains on the QR TCB until something (perhaps another DB2 call) forces it back to the open TCB once more.
    - A CICSAPI program only uses CICS APIs. Applications can run successfully in user key or CICS key irrespective of the key of the TCB. So they can run on the QR TCB, an L8 or an L9 TCB.
  - An OPENAPI THREADSAFE program always runs on an open TCB, and does so from the start of the program.
    - If use of a non threadsafe CICS command forces a switch to QR TCB, then CICS switches back to the open TCB again before returning control to the application.
    - An OPENAPI program can potentially use other (non CICS) APIs, and such APIs generally require the key of the TCB to match the execution key. Therefore user key programs run on L9 TCBs and CICS key programs run on L8 TCBs.



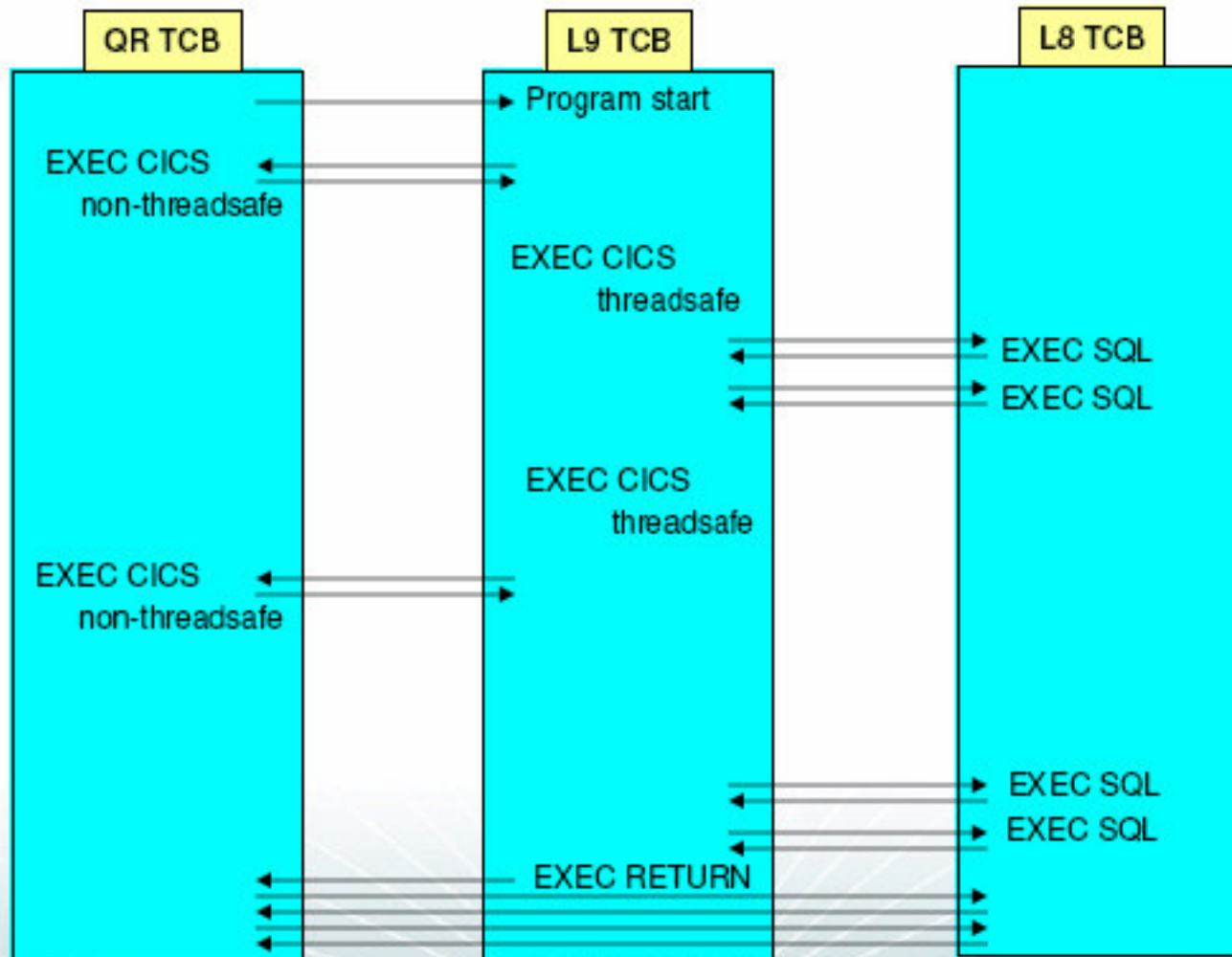
## CICS 3.1 TCB Switching



The program for transaction **BLUE** is defined **THREADSAFE**, API= CICSAPI

The program for transaction **RED** is defined **THREADSAFE**, API= OPENAPI, EXECKEY=CICS

## User Key OPEN API CICS – DB2 Application



## OPENAPI Notes

The use of OPENAPI programs allows application workloads to be moved off the QR TCB onto multiple open TCBs.

If you choose to use OPENAPI programs as a way of running workloads using other (non CICS) APIs remember that the **use of other (non CICS) APIs within CICS is entirely at the discretion and risk of the user. No testing of other (non CICS) APIs within CICS has been undertaken and use of such APIs is not supported by IBM Service.**

A new keyword (API) on the PROGRAM resource definition which takes one of two values CICSAPI or OPENAPI, where CICSAPI is the default.

A setting of API(OPENAPI) mandates a setting of CONCURRENCY(THREADSAFE) meaning the application must be coded to threadsafe standards so its application logic is capable of executing with integrity when executed in parallel on multiple TCBs. CICS will handle the threadsafety aspects of any CICS APIs issued from such programs.

The new program option applies to user application programs, PLT programs, user replaceable modules and task related user exits. It is ignored for global user exits.

## Program Definitions for Threadsafe Pre 3.1

- CONCURRENCY attribute is specified on the PROGRAM definition
  - {QUASIRENT | THREADSAFE}
  - Specifies whether the program is written to threadsafe standards or is only quasi-reentrant
  - Applies to:
    - User application programs
    - PLT programs
    - User-replaceable programs
    - Global user exit programs
    - Task-related user exit programs
  - CONCURRENCY can be specified in a program autoinstall exit
    - IBM-supplied sample (DFHPGAPG) specifies QUASIRENT

# Program Definitions for Threadsafe Pre 3.1

```
DEF PROG(THRDSAFE) GROUP(ADDISON)
```

```
OVERTYPE TO MODIFY
```

CICS RELEASE = 0620

```
CEDA DEFINE PROGRAM(THRDSAFE)
```

```
PROGRAM      : THRDSAFE
```

```
GROUP       : ADDISON
```

```
DESCRIPTION ==>
```

```
LANGUAGE    ==> CObol | Assembler | Le370 | C | Pli
```

```
RELOAD     ==> No | Yes
```

```
RESIDENT   ==> No | Yes
```

```
USAGE      ==> Normal | Transient
```

```
USELPACOPY ==> No | Yes
```

```
STATUS     ==> Enabled | Disabled
```

```
RS1        : 00 | 0-24 | Public
```

```
CEDF       ==> Yes | No
```

```
DATALOCATION ==> Below | Any
```

```
EXECKEY    ==> User | Cics
```

```
Concurrency ==> Quasirent | Threadsafe
```



## Program Definitions for Threadsafe 3.1

- API attribute is specified on the PROGRAM definition
  - {CICSAPI | OPENAPI}
  - Specifies if the program is limited to the CICS API or if it can execute in a threadsafe open environment
    - CICSAPI - the program is restricted to the CICS API
      - Execution is the same as R2.2 and R2.3
    - OPENAPI - the program execution begins on an OPEN TCB
      - The program is not restricted to the CICS API
  - API=OPENAPI requires CONCURRENCY(THREADSAFE)
  - Applies to:
    - User application programs
    - PLT programs
    - User-replaceable programs
    - Task-related user exit programs
  - OPENAPI can be specified in a program autoinstall exit
    - IBM-supplied sample (DFHPGAPG) specifies CICSAPI

# Program Definitions for Threadsafe 3.1

**CICS RELEASE = 640**

```

DEF PROG(THRDSAFE) GROUP(ADDISON)
OVERTYPE TO MODIFY
  CEDA DEFINE PROGRAM( THRDSAFE )
    PROGRAM      : THRDSAFE
    Group        : ADDISON
    Description   ==>
    Language     ==> CObol | Assembler | Le370 | C | Pli
    REload       ==> No      No | Yes
    RESident     ==> No      No | Yes
    USAge        ==> Normal  Normal | Transient
    USElpacopy   ==> No      No | Yes
    Status       ==> Enabled  Enabled | Disabled
    RS1          : 00        0-24 | Public
    CEdf         ==> Yes     Yes | No
    DAtalocation ==> Below   Below | Any
    EXECKey      ==> User    User | Cics
    Concurrency ==> Quasirent  Quasirent | Threadsafe
    Api         ==> Cicsapi    Cicsapi | Openapi
  
```

## Program Definition Notes

It's very important to understand the program definition keyword CONCURRENCY(THREADSAFE) is telling CICS the application logic is threadsafe, not whether CICS commands are threadsafe. CICS will ensure threadsafety of CICS API commands. If the command is not threadsafe, execution will be switched to the QR TCB. If the command is threadsafe, execution may remain on the open TCB. In either case, the resource is accessed in a threadsafe manner.

A threadsafe application can use non-threadsafe CICS commands. It will suffer the overhead of TCB switching but resource integrity is maintained.

If a non-threadsafe application is incorrectly defined to CICS as CONCURRENCY(THREADSAFE), the results are unpredictable.

The API parameter was added to the program definition in CICS Transaction Server R3.1. The default is API(CICSAPI) indicating the program is given control in the same manner as prior releases. Specifying API(OPENAPI) indicates CICS is to give control to the program on an OPEN (L8/L9) TCB. API(OPENAPI) requires the CONCURRENCY setting be specified as THREADSAFE.

### CICSAPI

The program is restricted to use of the CICS permitted application programming interfaces only. Dependent upon the program's CONCURRENCY setting, the application will either always run on the quasi-reentrant (QR) TCB, or if it is defined as threadsafe it may run on whichever TCB in use by CICS at the time which is determined as suitable.

### OPENAPI

The program is not restricted to the CICS permitted application program interfaces only. CICS will execute the program on its own L8 or L9 mode open TCB dependent upon the EXECKEY setting. If when executing a CICS command, CICS requires a switch to QR TCB, it will return to the open TCB before handing control back to the application program. OPENAPI requires the program to be coded to threadsafe standards and defined with CONCURRENCY(THREADSAFE).

It's important to note, API(OPENAPI) is ignored for GLUEs (Global User Exits).

## TRUE Definitions for Threadsafe

- BASEAPI (CICSAPI in R3.1)
  - TRUE enabled as either QUASIRENT or THREADSAFE
    - Without the OPENAPI option
    - Restricted to the CICS permitted programming interfaces
- OPENAPI
  - TRUE enabled with the OPENAPI option
    - Permitted to use non-CICS API
    - CICS will give control under an L8 mode open TCB
    - Assumes the TRUE is written to threadsafe standards
    - The L8 TCB is dedicated to the allocated task
    - Refer to the CICS Customization Guide for guidelines

## Threadsafe API Commands

### CICS TS 2.2

- ABEND
- ADDRESS
- ASSIGN
- DELETEQ TS
- DEQ
- ENQ
- ENTER TRACENUM
- FREEMAIN
- GETMAIN
- HANDLE ABEND
- HANDLE AID
- HANDLE CONDITION
- IGNORE CONDITION
- LINK
- LOAD
- MONITOR
- POP HANDLE
- PUSH HANDLE
- READQ TS
- RELEASE
- RETURN
- SUSPEND
- WAIT EXTERNAL
- WRITEQ TS
- XCTL

**Note: Wait External is Threadsafe, Wait Event is not!**



## Threadsafe API Commands

### CICS TS 2.3

- ASKTIME
- CHANGE TASK
- DOCUMENT CREATE
- DOCUMENT INSERT
- DOCUMENT RETRIEVE
- DOCUMENT SET
- FORMATTIME

# Threadsafe API Commands

## CICS TS 3.1 New Commands

- CONVERTTIME
- DELETE CONTAINER (CHANNEL)
- GET CONTAINER (CHANNEL)
- INVOKE WEBSERVICE
- MOVE CONTAINER (CHANNEL)
- PUT CONTAINER (CHANNEL)
- SOAPFAULT ADD
- SOAPFAULT CREATE
- SOAPFAULT DELETE
- WEB CONVERSE
- WEB CLOSE
- WEB OPEN
- WEB PARSE URL
- WEB RECEIVE (Client)
- WEB SEND (Client)

## CICS TS 3.1 Existing Commands

- WEB ENDBROWSE FORMFIELD
- WEB ENDBROWSE HTTPHEADER
- WEB EXTRACT
- WEB READ FORMFIELD
- WEB READ HTTPHEADER
- WEB READNEXT FORMFIELD
- WEB READNEXT HTTPHEADER
- WEB RECEIVE (Server)
- WEB RETRIEVE
- WEB SEND (Server)
- WEB STARTBROWSE FORMFIELD
- WEB STARTBROWSE HTTPHEADER
- WEB WRITE HTTPHEADER

## Threadsafe SPI Commands

### CICS TS 2.2

- DISCARD DB2CONN
- DISCARD DB2ENTRY
- DISCARD DB2TRAN
- INQUIRE DB2CONN
- INQUIRE DB2ENTRY
- INQUIRE DB2TRAN
- INQUIRE EXITPROGRAM
- INQUIRE TASK
- SET DB2CONN
- SET DB2ENTRY
- SET DB2TRAN

### CICS TS 2.3

- INQUIRE WORKREQUEST
- SET WORKREQUEST
- INQUIRE DOCTEMPLATE
- DISCARD DOCTEMPLATE

Note: Any new CICS TS 3.1 SPI Commands will be threadsafe

## Threadsafe XPI Commands

- All XPI commands are threadsafe except
  - DFHDUDUX TRANSACTION\_DUMP
  - DFHJCJCX WRITE\_JOURNAL\_DATA

## Steps for Ensuring Threadsafe - DFHEISUP

### ■ DFHEISUP

- IBM-supplied sample program provided in SDFHSAMP as DFHEILMS
- Sample filters also are provided in SDFHSAMP

### ■ Filter DFHEIDTH

- As provided, looks for CICS commands that give access to shared storage
  - EXTRACT EXIT
  - GETMAIN SHARED
  - ADDRESS CWA
- Can be modified to look for other CICS commands

### ■ Filter DFHEIDNT

- Contains the filter set for CICS commands that are not threadsafe and will cause a switch to the QR TCB



## Steps for Ensuring Threadsafe - DFHEISUP

- DFHEISUP can provide a summary report of the modules scanned
  - Summary report can be used as input for a detailed report

```
//DFHSCAN EXEC PGM=DFHEISUP, PARM=( 'SUMMARY, DETAILMODS' ), REGION=512M
//*****
//*      Sample JCL for running the Load Module Scanner          *
//*      The CICS Operations and Utilities Guide contains a detailed *
//*      description of the Load Module Scanner                  *
//**
//*****
//STEPLIB DD DSN=CTS220.CICS620.SDFHLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//DFHFLTR DD DSN=CTS220.CICS620.SDFHSAMP(DFHEIDTH), DISP=SHR
//DFHDTL DD DSN=USASSC1.MODL, DISP=OLD
//DFHIN DD DSN=USASSC1.CICSR620.P1LIB, DISP=SHR
```

**Note:** Program names found will be put into dataset named on DFHDTL  
Programs scanned are in dataset named on DFHIN

## Steps for Ensuring Threadsafe - DFHEISUP

- Output from DFHEISUP 'Summary,DetailMods'

```
CICS LOAD MODULE SCANNER UTILITY
SCAN PERFORMED ON Tue Jan 11 19:58:38 2005 USING TABLE
RSTABLE2.2

SUMMARY LISTING OF USASSC1.CICSR620.P1LIB
=====
Module Name      Commands Found  Language
'USASSC1.CICSR620.P1LIB(SQLASM) '      1  Assembler
'USASSC1.CICSR620.P1LIB(SQLASM1) '     1  Assembler
'USASSC1.CICSR620.P1LIB(SQLASM2) '     1  Assembler

LOAD LIBRARY STATISTICS
=====
Total modules in library                =      349
Total modules Scanned                   =      349
Total CICS modules/tables not scanned   =         0
Total modules possibly containing requested commands =         3
```

## Steps for Ensuring Threadsafe - DFHEISUP

- Use the output from DFHEISUP 'Summary,DetailMods' as input for Detail Report

```
//DFHSCAN EXEC PGM=DFHEISUP, PARM=('DETAIL'), REGION=512M
//*****
//*
//* Sample JCL for running the Load Module Scanner *
//*
//* The CICS Operations and Utilities Guide contains a detailed *
//* description of the Load Module Scanner *
//*****
//STEPLIB DD DSN=CTS220.CICS620.SDFHLOAD, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//DFHFLTR DD DSN=CTS220.CICS620.SDFHSAMP(DFHEIDTH), DISP=SHR
//DFHLIST DD DSN=USASSC1.MODL, DISP=SHR
//DFHIN DD DSN=USASSC1.CICSR620.P1LIB, DISP=SHR
```

**Note:** DFHLIST dataset contains programs found by 'Summary,Detailmods'

You can use PARM=('DETAIL,ALL') without DD DFHLIST to create Detail Report for all programs contained in DFHIN dataset

# Steps for Ensuring Threadsafe - DFHEISUP

- Output from DFHEISUP 'Detail'

```
CICS LOAD MODULE SCANNER UTILITY
SCAN PERFORMED ON Tue Jan 11 20:17:29 2005 USING Table RSTABLE2.2
DETAILED LISTING OF DD:DFHLIST
```

```
=====
Module Name      'USASSC1.CICSR620.P1LIB(SQLASM)'
Module Language  Assembler
Offset/EDF      Command
```

```
-----
00000934/no-edf  ADDRESS CWA
```

```
Module Name      'USASSC1.CICSR620.P1LIB(SQLASM1) '
Module Language  Assembler
Offset/EDF      Command
```

```
-----
00001028/no-edf  ADDRESS CWA
```

```
Module Name      'USASSC1.CICSR620.P1LIB(SQLASM2) '
Module Language  Assembler
Offset/EDF      Command
```

```
-----
00001036/no-edf  ADDRESS CWA
```

```
Total possible commands located = 3
```

```
LOAD LIBRARY STATISTICS
```

```
=====
Total modules in library           =      3
Total modules Scanned              =      3
Total CICS modules/tables not scanned =      0
Total modules possibly containing requested commands =      3
```

## DFHEISUP APARs

- **PQ73890** - DFHEISUP does not list EXEC CICS SEND MAP when the command contains the option MAPONLY
- **PQ76545** - ABEND 0C4 in module DFHEISUP scanning application load libraries
- **PQ77185** - CEE3204S the system detected a protection exception (SYSTEM COMPLETION CODE=0C4).
- **PQ78531** - Fixes a storage leak and the use of PDS with large number of members
- **PQ82603** - Running DFHEISUP returns an undocumented error message. A new table (DFHEIDAL) is introduced which contains all commands and DFHEIDNT which lists all non threadsafe commands.
- **PQ87863** - ASKTIME ABSTIME listed as non-threadsafe in filter DFHEIDNT
- **PQ99113** - DFHEISUP with parameter PARM=('DETAIL, ALL') displays offsets beyond the end of the programs



## Steps for Ensuring Threadsafe – RENTPGM=PROTECT

- DFHSIT Parameter RENTPGM=PROTECT

- Specifies you want CICS to allocate the read-only DSAs ( RDSA and ERDSA) from read-only key-0 protected storage
  - RDSA for RMODE(24) programs and the ERDSA for RMODE(ANY) programs
- Any attempt to overwrite the program will result in message DFHSR0622 and an ABEND0C4
  - Programs running in key-zero or supervisor state can still overlay RDSA and ERDSA

**DFHSR0622** An attempt to overwrite the RDSA has caused the abend which follows

**DFHAP0001** An abend (code 0C4/AKEA) has occurred at offset X'00000ACC' in module SQLSPIN.

## Checking Concurrency – DFH0STAT

Install Group DFH\$STAT from DFHCSD and run Transaction STAT

```
Sample Program - CICS Statistics Print          01/29/2005  14:21:14
Type in destination fields if required. Press Enter to print

Jobname. . . . : IYNXV
Applid . . . . : IYNXV
Sysid. . . . . : ISC1

Node . . . . . *           Type in a valid Node. * is default
Userid . . . . . *         Type in a valid Userid. * is default
Class. . . . . A          Type in a valid Class. A is default

Abbreviated. . . B        Type U or N for abbreviated report. B is default

Current Statistics Settings

Statistics Recording. : ON           Collection Interval . . . : 03:00:00
Last Reset Time . . . : 12:00:00     Elapsed Time Since Reset. : 02:21:14
Next Collection . . . : 15:00:00     End-of-Day Time . . . . . : 00:00:00

F1=Help F2=Refresh F3=Exit F4=Report Selection F5=Print
```

# Checking Concurrency – DFH0STAT

Ensure you select DB2 Connection and Entries  
 Ensure you select User Exits and Global User Exits

Sample Program - CICS Statistics Print Report Selection 01/29/2005 14:21:39

Select the statistics reports required and press 'Enter' to validate

<u>DB2 Connection</u> . . . . .	<u>Y</u>	<u>DB2 Entries</u> . . . . .	<u>Y</u>
Program Autoinstall. . . . .	N	Terminal Autoinstall and VTAM. . .	N
Connections and Modenames. . . . .	N		
Enqueue Manager. . . . .	N	Enqueue Models . . . . .	N
Recovery Manager . . . . .	N		
<u>User Exit Programs</u> . . . . .	<u>Y</u>	<u>Global User Exits</u> . . . . .	<u>Y</u>

F1=Help    **F3=Return to Print**    F7=Back    F10=Save    F12=Restore

## Checking Concurrency – DFH0STAT

Sample Program - CICS Statistics Print 01/29/2005 14:22:43

Type in destination fields if required. Press Enter to print

Jobname. . . : IYNXV  
Applid . . . : IYNXV  
Sysid. . . . : ISC1

Node . . . . . \* Type in a valid Node. \* is default  
Userid . . . . . \* Type in a valid Userid. \* is default  
Class. . . . . A Type in a valid Class. A is default

Abbreviated. . B Type U or N for abbreviated report. B is default

### Current Statistics Settings

Statistics Recording. : ON	Collection Interval . . . : 03:00:00
Last Reset Time . . . : 15:00:00	Elapsed Time Since Reset. : 00:02:42
Next Collection . . . : 18:00:00	End-of-Day Time . . . . . : 00:00:00

**Statistics print successfully completed**

F1=Help F2=Refresh F3=Exit F4=Report Selection **F5=Print**

# Checking Concurrency – DFH0STAT

## User Exit Report

### User Exit Programs

Program Name	Entry Name	Entry Name	Length	Use Count	No. of Exits	Program Status	<u>Program Concurrency</u>	Exit Program Use Count
DFHEDP	DLI		0	0	0	Started	Quasi Rent	0
DFHD2EX1	DSNCSQL	DSNCSQL	16	1	0	Started	Quasi Rent	234
EDZEXIT	EDZEXIT		0	0	1	Started	Thread Safe	234
EDZEXIT2	EDZEXIT2	<b>EDZEXIT2</b>	<b>64</b>	1	1	Started	Thread Safe	234

Applid IYNXV      Sysid ISC1    Jobname IYNXV      Date 01/28/2005    Time 14:22:43      CICS 6.2.0

Program Name	Entry Name	API	Concurrency Status	Qualifier	Length	Taskstart	EDF	Shutdown	Indoubt	SPI	Purgeable
DFHEDP	DLI	Base	Quasi Rent		284	No	No	No	No Wait	No	No
DFHD2EX1	DSNCSQL	Open	Thread Safe	DF23	222	No	Yes	Yes	Wait	Yes	Yes
EDZEXIT	EDZEXIT	Base	Thread Safe		0	No	No	No	No Wait	No	No
EDZEXIT2	EDZEXIT2	Base	Thread Safe		0	No	No	No	No Wait	No	No

Applid IYNXV      Sysid ISC1    Jobname IYNXV      Date 01/29/2005    Time 14:22:43      CICS 6.2.0

Note: EDZEXIT2 is defined as threadsafe and utilizes a Global Workarea



# Checking Concurrency – DFH0STAT Notes

## Field Heading Description

Program Name	The program name of the program that has been enabled as an exit program using the EXEC CICS ENABLE command	
Entry Name	The entry point name for this exit program	
Global Area Entry Name	The name of the exit program that owns the global work area associated with this exit program	
Global Area Length	The length of the global work area associated with this exit program	
Global Area Use Count	The number of exit programs that are associated with the global work area owned by this exit program	
Number of Exits	The number of global user exit points at which this exit program is enabled	
Program Status	Indicates whether this exit program is available for execution	
Program Concurrency	Indicates the concurrency attribute of this exit program	
Exit Program Use Count	The number of times this exit program has been invoked	
Program Name	The program name of the program that has been enabled as an exit program using the EXEC CICS ENABLE command	
Entry Name	The entry point name for this exit program	
API	Indicates which APIs the task-related user exit program uses. BASEAPI or OPENAPI	
Concurrency Status	Indicates the concurrency attribute of the exit program. The values are Quasirent or Threadsafe	
Qualifier	The name of the qualifier specified for this exit program	
Length	The length of the task local work area associated with this exit program	
Task Related User Exit Options – Taskstart.	Indicates whether this exit program was enabled with the TASKSTART option	
Task Related User Exit Options – EDF	Indicates whether this exit program was enabled with the FORMATEDF option	
Task Related User Exit Options – Shutdown	Indicates whether this exit program was enabled with the SHUTDOWN option	
Task Related User Exit Options – Indoubt	Indicates whether this exit program was enabled with the INDOUBTWAIT option	
Task Related User Exit Options – SPI	Indicates whether this exit program was enabled with the SPI option	
Task Related User Exit Options – Purgeable	Indicates whether this exit program was enabled with the PURGEABLE option	

# Checking Concurrency – DFH0STAT

## Global User Exit Report

### Global User Exits

Exit Name	Program Name	Entry Name	<----- Global Area ----->		Number of Exits	Program Status	<u>Program Concurrency</u>
			Entry Name	Length	Use Count		
XRMIIN	EDZEXIT	EDZEXIT		0	0	1 Started	Thread Safe
XRMIIN	EDZEXIT2	EDZEXIT2		0	0	1 Started	Quasi Rent
XRMOUT	EDZEXIT	EDZEXIT		0	0	1 Started	Thread Safe
XRMOUT	EDZEXIT2	EDZEXIT2		0	0	1 Started	Quasi Rent

## Checking Concurrency – DFH0STAT Notes

### Field Heading Description

Exit Name	The name of the global user exit point
Program Name	The name of the exit program enabled at this global user exit point
Entry Name	The name of the entry point for this exit program at this global user exit point
Global Area Entry Name	The name of the exit program that owns the global work area associated with this exit program
Global Area Length	The length of the global work area for this exit program
Global Area Use Count	The number of exit programs associated with the global work area owned by this exit program
Number of Exits	The number of global user exit points at which this exit program is enabled
Program Status	Indicates whether this exit program is available for execution
Program Concurrency	Indicates the concurrency attribute of this exit program

# Checking Concurrency – DFH0STAT

## DB2 Connection Report

DB2 Connection

```

DB2 Connection Name. . . . . : DF23
DB2 Group Id . . . . . :
DB2 Sysid. . . . . : DF23
DB2 Release. . . . . : 7.1.0
DB2 Connection Status. . . . . : CONNECTED
DB2 Connection Error . . . . . : SQLCODE
DB2 Standby Mode . . . . . : RECONNECT
DB2 Pool Thread Plan Name. . . . . :
DB2 Pool Thread Dynamic Plan Exit Name . . . . . : DSNCUEXT
Pool Thread Authtype . . . . . : USERID
Pool Thread Authid . . . . . :
Signid for Pool/Entry/Command Threads. . . : IYNXV
Create Thread Error. . . . . : N906D
Protected Thread Purge Cycle . . . . . : 00.30
Deadlock Resolution. . . . . : ROLLBACK
Non-Terminal Intermediate Syncpoint. . . : RELEASE
Pool Thread Wait Setting . . . . . : WAIT
Pool Thread Priority . . . . . : HIGH
Current TCB Limit. . . . . : 12
Pool Thread Limit. . . . . : 3
Current number of Pool Threads . . . . . : 0
Peak number of Pool Threads. . . . . : 0
Number of Pool Thread Waits. . . . . : 0

Current number of Pool Tasks . . . . . : 0
Peak number of Pool Tasks. . . . . : 0
Current Total number of Pool Tasks . . . : 0
Current number of Tasks on Pool Readyq . : 0
Peak number of Tasks on Pool Readyq. . . : 0
Current number of DSN Command threads . . : 0
Peak number of DSN Command threads. . . : 0
DSNC Command Thread Limit. . . . . : 1

Resync Group Member . . . . . : N/A
DB2 Connect Date/Time: 01/28/2005 15:18:04

Command Thread Authtype . . . . . : USERID
Command Thread Authid . . . . . :
Message TD Queue 1. . . . . : CDB2
Message TD Queue 2. . . . . :
Message TD Queue 3. . . . . :
Statistics TD Queue . . . . . : CDB2
DB2 Accounting records by . . . . . : NONE

Number of Calls using Pool Threads. . . : 0
Number of Pool Thread Signons . . . . . : 0
Number of Pool Thread Partial Signons . : 0
Number of Pool Thread Commits . . . . . : 0
Number of Pool Thread Aborts. . . . . : 0
Number of Pool Thread Single Phase. . . : 0
Number of Pool Thread Reuses. . . . . : 0
Number of Pool Thread Terminates. . . : 0

Number of DSN Command Calls. . . . . : 0
Number of DSN Command Signons. . . . . : 0
Number of DSN Command Thread Terminates: 0
    
```

# Checking Concurrency – CEDA

## CEDA V PROGRAM(DSNCUEXT)

```

OBJECT CHARACTERISTICS                                CICS RELEASE = 0620
CEDA View PROGram( DSNCUEXT )
PROGRAM      : DSNCUEXT
Group       : DFHCOMP6
DEscription : CICS-DB2 DYNAMIC PLAN SELECTION EXIT
Language    : Assembler          CObol | Assembler | Le370 | C | Pli
RELoad     : No                  No | Yes
RESident   : No                  No | Yes
USAge      : Normal              Normal | Transient
USElpacopy : No                  No | Yes
Status     : Enabled             Enabled | Disabled
RS1        : 00                  0-24 | Public
CEdf       : No                  Yes | No
DATalocation : Below             Below | Any
EXECKey    : Cics                User | Cics
COncurrency : Quasirent        Quasirent | Threadsafe
REMOTE ATTRIBUTES
Dynamic    : No                  No | Yes
+ REMOTESystem :

```

Note: PQ67351 provided Threadsafe Dynamic Plan Exit DFHD2PXT



# Checking Concurrency – DFH0STAT

## DB2 Entries Report

DB2 Entries

DB2Entry Name. . . . .	: CWA1	DB2Entry Status . . . . .	: ENABLED
DB2Entry Static Plan Name. . . . .	:	DB2Entry Disabled Action. . . . .	: POOL
<b><u>DB2Entry Dynamic Plan Exit Name. . . . .</u></b>	<b>: <u>SQLASM1</u></b>	DB2Entry Deadlock Resolution. . . . .	: ROLLBACK
DB2Entry Authtype. . . . .	: N/A	DB2Entry Accounting records by. . . . .	: NONE
DB2Entry Authid. . . . .	: USASSC1		
		Number of Calls using DB2Entry. . . . .	: 0
DB2Entry Thread Wait Setting . . . . .	: POOL	Number of DB2Entry Signons. . . . .	: 0
		Number of DB2Entry Partial Signons. . . . .	: 0
DB2Entry Thread Priority . . . . .	: HIGH	Number of DB2Entry Commits. . . . .	: 0
DB2Entry Thread Limit. . . . .	: 0	Number of DB2Entry Aborts . . . . .	: 0
Current number of DB2Entry Threads . . . . .	: 0	Number of DB2Entry Single Phase . . . . .	: 0
Peak number of DB2Entry Threads. . . . .	: 0	Number of DB2Entry Thread Reuses. . . . .	: 0
		Number of DB2Entry Thread Terminates. . . . .	: 0
DB2Entry Protected Thread Limit. . . . .	: 0	Number of DB2Entry Thread Waits/Overflows: 0	
Current number of DB2Entry Protected Threads . . . . .	: 0		
Peak number of DB2Entry Protected Threads. . . . .	: 0		
Current number of DB2Entry Tasks . . . . .	: 0		
Peak number of DB2Entry Tasks. . . . .	: 0		
Current Total number of DB2Entry Tasks . . . . .	: 0		
Current number of Tasks on DB2Entry Readyq . . . . .	: 0		
Peak number of Tasks on DB2Entry Readyq. . . . .	: 0		

## Checking Concurrency - CEMT

```
I PROGRAM(SQLASM1)  
STATUS: RESULTS - OVERTYPE TO MODIFY  
Prog(SQLASM1 ) Leng(0000000744)      Pro Ena Pri      Ced  
Res(000) Use(0000000000) Bel Uex Ful Thr
```

RESPONSE: NORMAL

SYSID=ISC1 APPLID=IYNXV  
TIME: 15.22.32 DATE: 01.02.05

## Threadsafe Problem One

**DFHPG0001** An abend (code 0C1/AKEA) has occurred at offset X'FFFF' in module DFHPGPG

DFHME0116 (Module:DFHMEME) CICS symptom string for message DFHPG0001 is  
PIDS/5697E9300 LVL5/620 MS/DFHPG0001 RIDS/DFHPGPG PTFS/UQ94391  
AB/S00C1 AB/UAKEA ADRS/0000FFFF

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: PG0001 , DUMPID: 1/0043

**DFHXM0001** An abend (code ---/APGB) has occurred at offset X'0D4A' in module DFHXMTA

DFHME0116 (Module:DFHMEME) CICS symptom string for message DFHXM0001 is  
PIDS/5697E9300 LVL5/620 MS/DFHXM0001 RIDS/DFHXMTA PTFS/UQ85452  
AB/UAPGB ADRS/00000D4A

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: XM0001 , DUMPID: 1/0044

**DFHAP0001** An abend (code 0C1/AKEA) has occurred at offset X'000007E0' in module **SQLSPIN**

DFHME0116 (Module:DFHMEME) CICS symptom string for message DFHAP0001 is  
PIDS/5697E9300 LVL5/620 MS/DFHAP0001 RIDS/DFHSRP PTFS/UQ83467  
AB/S00C1 AB/UAKEA RIDS/SQLSPIN ADRS/000007E0

## Threadsafe Problem One

**DFHAP0001** An abend (code 0C6/AKEA) has occurred at offset X'FFFF' in module DFHERMSP

DFHME0116 (Module:DFHMEME) CICS symptom string for message DFHAP0001 is  
PIDS/5697E9300 LVLS/620 MS/DFHAP0001 RIDS/DFHERMSP PTFS/UQ79266

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: AP0001 , DUMPID: 1/0045  
AB/S00C6 AB/UAKEA RIDS/DFHERMSP ADRS/0000FFFF

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: AP0001 , DUMPID: 1/0046

**DFHRM0002** A severe error (code X'0367') has occurred in module DFHRMLSD

DFHDU0201 ABOUT TO TAKE SDUMP. DUMPCODE: RM0002 , DUMPID: 1/0047

BPXP018I THREAD 13C7110000000003, IN PROCESS 50331679, ENDED  
WITHOUT BEING UNDUBBED WITH COMPLETION CODE 0033E000,  
AND REASON CODE 00000000.

DFHDU0303I Transaction Dump Data set DFHDMPA closed

**DFHKE1800 ABNORMAL TERMINATION OF CICS IS COMPLETE**

# Threadsafe Problem One – KE=3

```

=== DUMP SUMMARY

DUMPID:      1/0043

DUMPCODE:   PG0001

DATE/TIME:  20/12/04 15:55:54 (LOCAL)

MESSAGE:    DFHPG0001 An abend (code 0C1/AKEA) has occurred at offset X'FFFF' in module DFHPGPG
    
```

===KE: Kernel Domain KE\_TASK Summary

KE_NUM	KE_TASK	STATUS	TCA_ADDR	TRAN_#	TRANSID	DS_TASK	KE_KTCB	ERROR
0400	161E4780	Unused						
0401	161E4B00	***Running**	00045080	00387	SQLS	3319C880	13ADC020	
0402	161FB080	***Running**	0005B080	00381	SQLS	3319C780	13ADB020	
0403	161FB400	Not Running	0005A080	00380	SQLS	3319C680	13ADA020	
0404	161FB780	***Running**	0005B680	00382	SQLS	3319C580	13AD9020	
0405	161FBB00	***Running**	00042080	00383	SQLS	3319C480	13AD8020	
0406	161FC080	Not Running	00058680	00388	CEMT	3319C380	13A99020	
<b>0407</b>	<b>161FC400</b>	<b>***Running**</b>	<b>00044080</b>	<b>00385</b>	<b>SQLS</b>	<b>3319C080</b>	<b>13A99020</b>	<b>*YES*</b>
0408	161FC780	***Running**	00044680	00386	SQLS	3319C180	33021020	
0409	161FCB00	Not Running	00042680	00384	SQLS	3319C280	13A99020	
040B	1627D780	Not Running	13CB1680	00024	CEX2	3300AD80	13A99020	
040F	1629C400	Not Running	0005A680	00021	CSNC	330C7D80	13A99020	
0411	1629CB00	Not Running	13CB0680	00019	CFQS	3300AA80	13A99020	



## Threadsafe Problem One – KE=3

KE_NUM	@STACK	LEN	TYPE	ADDRESS	LINK	REG	OFFS	ERROR	NAME
0407	1622A020	0130	Bot	93901600	939018AC	02AC			DFHKETA
0407	1622A150	0260	Dom	93916618	9391671E	0106			DFHDSKE
0407	1622A3B0	0520	Dom	93937560	939382AA	0D4A			DFHXMTA
0407	1622A8D0	0600	Dom	93E7B7C0	93E7F5DC	3E1C		<b>*YES*</b>	DFHPGPG
			Int	+01FA	93E7B85C	009C			INITIAL_LINK
0407	1622B690	0F90	Dom	93976490	93979E16	3986			DFHMEME
			Int	+2EFC	93976612	0182			SEND
			Int	+153C	9397947A	2FEA			CONTINUE_SEND
			Int	+38AE	93977A66	15D6			TAKE_A_DUMP_FOR_CALLER
0407	1622C620	0620	Dom	93A041A0	93A04D1C	0B7C			DFHDUDU
			Int	+0A1E	93A04298	00F8			SYSTEM_DUMP
			Int	+1826	93A04FEC	0E4C			TAKE_SYSTEM_DUMP

## PG=3

PPTE	MOD	LANG	CEDF	DATA	DPL	LOAD	USE	
ADDRESS	TYPE	DED	STAT	LOC	SUBS	STAT	COUNT	
PROGRAM	LANG	INST	AVAL	EXEC	RE	HOLD		<b><u>THREAD</u></b>
NAME	DEF	TYPE	STAT	KEY	LOAD	STAT		<b><u>SAFE</u></b>
164B3818	SQLASM1	PG NDF	ASS R	CED E	B U	F N	L T	Y
164B3768	SQLASM2	PG NDF	ASS R	CED E	B U	F N	L T	Y
164B3870	<b><u>SQLSPIN</u></b>	PG NDF	CO2 R	CED E	B U	F N	L T	7 <b><u>Y</u></b>
1641FB30	STARTBR	PG ASS	NDD G	CED E	B U	F N	ND T	N
1641FC90	WRITE23	PG ASS	NDD G	CED E	B U	F N	ND T	N

## Threadsafe Problem One – COBOL Program SQLSPIN

```
MOVE 1 TO COUNTER.  
PERFORM WITH TEST BEFORE  
  UNTIL COUNTER > 20  
  EXEC SQL  
    SELECT * INTO :VVEMP FROM DSN8710.EMP  
    WHERE EMPNO = '000990'  
  END-EXEC  
  
  CALL 'ASMSETR' END-CALL  
  ADD 1 TO COUNTER  
END-PERFORM.  
MOVE 'TRANSACTION COMPLETED.' TO LOGMSG  
EXEC CICS SEND FROM(LOGMSG) LENGTH(22) ERASE END-EXEC.  
  
END-MAIN.
```

Note: Static and Dynamic Called programs will run on the current TCB  
ASMSETR is a Static Called program and will run on the L8 TCB

## Threadsafe Problem One – What we know so far

- CICS Region terminated with DFHKE1800
  - Prior ABEND0C1 in user program SQLSPIN
- Task 00385 transaction SQLS program SQLSPIN suffered the original abend
- SQLSPIN is a COBOL program defined as Threadsafe
- SQLSPIN Issued DB2 Select command and called Assembler program ASMSETR

## Threadsafe Problem One – Assembler Program

```
ASMSETR  CSECT
          USING *,15
RSA      DC    16F'00'          REGISTER SAVE AREA
BEGIN     DS    0H
          SAVE  (14,12)         SAVE R14 Through R12
          ST   13,RSA+4        STANDARD SAVE AREA FOR R13 RECEIVED
          LA   14,RSA          POINT R14 AT SAVE AREA WITHIN PROGRAM
          ST   14,8(,13)       POST FORWARD RSA POINTER
          LR   13,14          POINT R13 TO CURRENT RSA
          LR   3,15           SET BASE REGISTER 3
          USING ASMSETR,3
*         Loop a little bit      *
          L    8,=X'00400000'
LOOP      MVC  FILLER,0(13)
          BCT  8,LOOP          BRANCH BACK TO LOOP
          L    13,4(13)
          RETURN (14,12),T,RC=0  RETURN TO SQLSPIN
FILLER    DC   C'XXXXXXXX'
          END   ASMSETR
```

Note: ASMSETR is called on L8 TCB

# Threadsafe Problem One – TR=2

```

AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE (DSNCSQL )
  TASK-00387 KE_NUM-0401 TCB-L8007/008AF2F0 RET-500C878A TIME-15:55:43.9283438300           =013978=
AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE (DSNCSQL )
  TASK-00385 KE_NUM-0407 TCB-L8000/008B2420 RET-500C878A TIME-15:55:48.5520001574           =017643=
AP 2520 ERM ENTRY COBOL-APPLICATION-CALL-TO-TRUE (DSNCSQL )
  TASK-00387 KE_NUM-0401 TCB-L8007/008AF2F0 RET-500C878A TIME-15:55:49.8550418298           =018863=
AP 2521 ERM EXIT COBOL-APPLICATION-CALL-TO-TRUE (DSNCSQL )
  TASK-00387 KE_NUM-0401 TCB-L8007/008AF2F0 RET-500C878A TIME-15:55:49.8552940485           =018868=
AP 1949 APLI EVENT RETURN-FROM-LE/370 - Rununit_Init_&_Begin_Invo OK Program_name (SQLSPIN)
  TASK-00387 KE_NUM-0401 TCB-L8007/008B2420 RET-93E7C78E TIME-15:55:54.4162750329           =023132=
AP 1948 APLI EVENT CALL-TO-LE/370 - Rununit_End_Invocation      Program_name (SQLSPIN)
  TASK-00387 KE_NUM-0401 TCB-L8007/008B2420 RET-93E7C78E TIME-15:55:54.4162841267           =023133=
AP 1949 APLI EVENT RETURN-FROM-LE/370 - Rununit_End_Invocation      OK Program_name (SQLSPIN)
  TASK-00387 KE_NUM-0401 TCB-L8007/008B2420 RET-93E7C78E TIME-15:55:54.4162985173           =023134=
AP 1948 APLI EVENT CALL-TO-LE/370 - Rununit_Termination          Program_name (SQLSPIN)
  TASK-00387 KE_NUM-0401 TCB-L8007/008B2420 RET-93E7C78E TIME-15:55:54.4163026267           =023135=
AP 00E1 EIP ENTRY FREEMAIN          REQ(0004) FIELD-A(16380060 ...-) FIELD-B(08000C04 .....)
  TASK-00385 KE_NUM-0407 TCB-L8000/008B2420 RET-8002B986 TIME-15:55:54.4163302048           =023136=
SM 0D01 SMMF  ENTRY - FUNCTION(FREEMAIN) ADDRESS(001C11C8) CALLER(EXEC) EXEC_KEY(CICS)
  TASK-00385 KE_NUM-0407 TCB-L8000/008B2420 RET-94351AD2 TIME-15:55:54.4163423454           =023137=
SM 0D0C SMMF  *EXC* - Invalid_freemain_address - FUNCTION(FREEMAIN) ADDRESS(001C11C8)
  TASK-00385 KE_NUM-0407 TCB-L8000/008B2420 RET-94351AD2 TIME-15:55:54.4163501267           =023138=
PG 0903 PGPG  *EXC* - Recovery - FUNCTION(INITIAL_LINK) PROGRAM_NAME (SQLSPIN)           =023658=
TASK-00385 KE_NUM-0407 TCB-QR      /008C9630 RET-939382AA TIME-15:55:54.5026572202
1-0000  00280000 0000009C 00000000 00000000  BC000000 00000000 01000100 E2D8D3E2  *.....SQLS*
   0020  D7C9D540 40404040                                *PIN
2-0000  F0C3F161 C1D2C5C1 018400C1 0000FFFF  C4C6C8D7 C7D7C740 13E7B7C0 3319C080  *OCI/AKEA.d.A....DFHPGPG .X.{..{*
  
```



## Threadsafe Problem One - What We Know

- CICS Region terminated with DFHKE1800
  - Prior ABEND0C1 in user program SQLSPIN
- Task 00385 transaction SQLS program SQLSPIN suffered the original abend
- SQLSPIN is a COBOL program defined as Threadsafe
- SQLSPIN Issued DB2 Select command and called Assembler program ASMSETR
- ASMSETR will run on the L8 TCB since it was invoked with a CALL
- ASMSETR stores R13 and calling registers within itself and loops
- Two L8 TCBs can both be in ASMSETR at the same time and pickup the wrong registers

## Threadsafe Problem One – Detection and Prevention

### ■ Detection

- This problem would have been detected if the load module containing the COBOL and Assembler program had been Link-Edited with the RENT attribute
  - Since ASMSETR stores within itself you would receive an ABEND0C4 when trying to update the Read Only DSA

### ■ Prevention

- Make the call to ASMSETR using EXEC CICS LINK or XCTL and ensure it is defined as Quasirent

Note: ASMSETR gains control by a static call so there is no need for a CICS RDO program definition, it is Link-Edited with the COBOL program SQLSPIN. Even if ASMSETR was a dynamic call program defined as Quasirent, CICS would not switch TCBs. This is due to CICS Only knowing about the original program.

## Threadsafe Problem Two

- Problem is described as yearly bonus totals for executives are being given to non-executives
- Program to figure bonuses has just been defined as threadsafe

# Threadsafe Problem Two - Program

```

DFHEISTG DSECT
EXEC SQL INCLUDE SQLCA
EMPNUM DS CL6
BONUSLNG DS OCL8
BONUSFIL DS CL5
BONUSPAK DS CL3
CWAREG EQU 9
SQDWSREG EQU 8
SQDWSTOR DS (SQLDLEN)C RESERVE STORAGE TO BE USED FOR SQLDSECT
CWAMAP DSECT
BONUSTOT DS XL4
SQLASM1 CSECT
*
MVC EMPNUM,=C'000140'
*
* SQL WORKING STORAGE
LA SQDWSREG,SQDWSTOR GET ADDRESS OF SQLDSECT
USING SQLDSECT,SQDWSREG AND TELL ASSEMBLER ABOUT IT
*
EXEC SQL
DECLARE DSN8710.EMP TABLE (
EMPNO CHAR(6),
SALARY DECIMAL,
BONUS DECIMAL,
*
EXEC SQL SELECT BONUS INTO :BONUS FROM DSN8710.EMP WHERE
EMPNO= :EMPNUM
*
XC BONUSLNG,BONUSLNG
MVC BONUSPAK,BONUS
*
EXEC CICS ADDRESS CWA(CWAREG)
USING CWAMAP,CWAREG
*
NEWTOTAL EQU 6
*
CVB NEWTOTAL,BONUSLNG Get current bonus in register.
A NEWTOTAL,BONUSTOT Add to old bonus total.
ST NEWTOTAL,BONUSTOT Update CWA with new total.
*
EXEC CICS RETURN
END

```

## Threadsafe Problem Two – Detection and Prevention

- Detection
  - DFHEISUP would have caught the EXEC CICS ADDRESS(CWA)
- Prevention
  - EXEC CICS ENQUEUE and DEQUEUE around the update to the CWA
  - Compare and Swap (CS) or Compare Double and Swap (CDS) for Assembler programs
  - Test and Set (TS) instruction for Assembler programs
  - Change RDO Program Definition to Quasirent



# Threadsafe Problem Three

- Problem is describes as number of transactions started does not match CICS statistics – Program for BIGG defined as OPENAPI

<pre> STARTER  CSECT EXEC CICS SET STATISTICS RECORDING(OFF) RESETNOW RESP(RSP) EXEC CICS SET STATISTICS RECORDING(ON) RESETNOW RESP(RSP) GETMAIN RC, LV=400, SP=10 ST      1, COMMADDR L       8, =X'00010000' STARTEM DS      0H EXEC CICS START TRANSID(BIGG) FROM(COMMADDR) LENGTH(HALF) BCT     8, STARTEM EXEC CICS SEND TEXT FROM(MESS1) LENGTH(MESS1LEN) MVC     MESSINLN, =X'0010' EXEC CICS RECEIVE INTO(MESSIN) LENGTH(MESSINLN) EXEC CICS COLLECT STATISTICS SET(8) TRANSACTION(BIGG)  *       USING DFHXMRDS, 8       MVC     TTR, XMRAC       UNPK   INDEXU, TTR(5)       TR     INDEX, HEXTAB       MVC     TRACK, INDEX  *       EXEC CICS SEND FROM(MESS2) LENGTH(MESS2LEN)       MVC     SBA, =X'11406D'       EXEC CICS SEND FROM(COUNTER)  *       EXEC CICS SEND FROM(MESS3) LENGTH(MESS3LEN)       L       8, COMMADDR       USING  COMMMap, 8       MVC     TTR, TRNCOUNT       UNPK   INDEXU, TTR(5)       TR     INDEX, HEXTAB       MVC     TRACK, INDEX       MVC     SBA, =X'11C17D'       EXEC CICS SEND FROM(COUNTER)       FREEMAIN RC, LV=400, A=(8), SP=10       EXEC CICS RETURN         </pre>	<pre> Reset Statistics Turn Statistics ON GETMAIN COMMAREA to pass  START BIGG Trans passing GETMAIN area COMMADDR will contain number of trans started Send message BIGG trans started  Get stats for transaction  Send message from stats with the number of attached BIGG transactions Append count to message  Send message from COMMADDR with the number of BIGG transactions that ran  Append count to message  FREEMAIN COMMAREA         </pre>
---	--

## Threadsafe Problem Three

- Transaction BIGG adds one to commarea and returns
- Commarea count versus EXEC CICS Statistic counts after BIGG transaction was started 65536 times

Program Logic for transaction BIGG	Commarea Count	CICS Statistic Count
SQL Select followed by DELAY INTERVAL(000001)	7620	65536
SQL Select followed by EXEC CICS SUSPEND	40080	65536
SQL Select followed by EXEC CICS ADDRESS EIB	52678	65536
SQL Select followed by 500 MVC Instructions	62335	65536
SQL Select followed by normal business logic	65534	65536

## Problem Three – Detection and Prevention

- No supplied way to detect the MVS Getmain of the storage used for the Commarea
  - DFHEISUP would not detect this MVS Macro
  - RENTPGM=PROTECT would not cause an Abend since the program is not storing the shared storage within itself
- Prevent by serializing access to COMMADDR in started transactions

## Summary

- Defining a program as threadsafe is not a guarantee the program logic is threadsafe
- Use supplied tools for initial pass at making your program threadsafe
  - DFHEISUP
  - RENTPGM=PROTECT
- Test your program under heavy loads to ensure desired results
- Enjoy the benefits of threadsafe and the Open Transaction Environment



# Questions and Answers