Understand how the new features in
CICS scalability can help your business

July 2011

IBM CICS scalability: New features in V4.2

John Tilling, Senior Developer, CICS Transaction Server

CICS icon of progress: http://www.ibm.com/ibm100/us/en/icons/cics/

Executive summary

CICS® Transaction Server is IBM®'s general-purpose transaction processing software for z/OS®. It is a powerful application server that meets the transaction-processing needs of both large and small enterprises. It builds on z/OS and System z® facilities to provide high availability and scalability at a low cost per transaction; it supports large transaction volumes with a fast and consistent response time.

CICS Transaction Server for z/OS handles billions of transactions a week. Companies around the world rely on their CICS systems in their daily operation. It is very important that the CICS systems are able to scale to meet the demands of the customer's business. CICS Transaction Sever is able to provide a scalable solution that scales horizontally by supporting the distribution and control of work between multiple CICS regions in a z/OS sysplex.

The newest release, CICS Transaction Server for z/OS Version 4.2, includes scalability enhancements that allow more work to be done more quickly in a single CICS system. This enhancement allows increased vertical scaling and may decrease the need to scale horizontally, thus reducing the number of regions required to run the production business applications.

The scalability enhancements in CICS Transaction Server Version 4.2 fall into two broad areas: increased exploitation of Open Transaction Environment (OTE) and increased exploitation of 64-bit storage.

Introduction to CICS OTE

CICS Open Transaction Environment (OTE) is an architecture that was introduced for three purposes:

- To allow CICS to make better use of the mainframe. OTE enables CICS to do more things in parallel, thereby increasing the throughput of work through the system. This results in more work being done in the same amount of time.
- To improve the performance of existing applications, particularly those that access external resources managers like DB2®, by consuming less mainframe resources in getting the job done.
- To augment the already rich set of capabilities provided by the CICS application programming interface, by providing application interfaces supplied by other software components, and allowing CICS applications to use these interfaces.

To benefit from the power of OTE, customers need to make sure that their applications are threadsafe. This ensures that if the mainframe has lots of CPUs and lots of processes are running in parallel, an application that is threadsafe runs correctly and the right result is achieved. CICS makes sure its code runs correctly, but customers need to make sure that their COBOL code, for example, implementing payroll, accounts, and ledger, runs correctly. If an application is threadsafe it can be defined to CICS by means of a CONCURRENCY keyword such that it uses OTE. If an application is not threadsafe, CICS runs it without using OTE.

Applications that cannot use OTE have to run on the main CICS TCB, the *QR TCB*, whereas applications that use OTE can run on a CICS open TCB. A CICS system has only one QR TCB, and the CICS dispatcher shares out use of the QR TCB between all the tasks. However a single CICS system can have many hundred open TCBs. Exploiting OTE effectively keeps an application running on an open TCB for as long as possible, and minimizes the number of times it has to switch back to the QR TCB. This provides CPU savings and improves throughput because the open TCBs can run in parallel and take advantage of the multiprocessor mainframe.

OTE enhancements in CICS TS 4.2 fall into three areas:

- The introduction of a new concurrency option on the program definition that allows for greater exploitation of OTE for threadsafe applications.
- Exploitation of OTE for function shipping, by allowing the mirror program, when it is invoked in a remote CICS region via an IPIC connection, to run on an open TCB.
- Making more of the application programming interface (API) and system programming interface (SPI) threadsafe, including access to IMS™ databases via the CICS-DBCTL interface.

Concurrency settings

Prior to CICS TS 4.2, an application program can be defined as CONCURRENCY(QUASIRENT) or CONCURRENCY(THREADSAFE).

- A CONCURRENCY(QUASIRENT) program always runs on the QR TCB.
- A CONCURRENCY(THREADSAFE) program is a program that has been coded to threadsafe standards and contains threadsafe logic. It is capable of running on either the QR TCB or an open TCB. It starts off running on the QR TCB. If processing such as a DB2 request causes a switch to an open TCB, then on return to the program the program continues on the open TCB.

CICS TS 4.2 provides a new CONCURRENCY(REQUIRED) setting. As with CONCURRENCY(THREADSAFE), the new setting specifies that the program has been coded to threadsafe standards and contains threadsafe logic, but in addition the program must run on an open TCB. Therefore the program runs on an open TCB from the start, and if CICS has to switch to the QR TCB to process a non-threadsafe CICS command, CICS returns to the open TCB when it returns control to the application program.

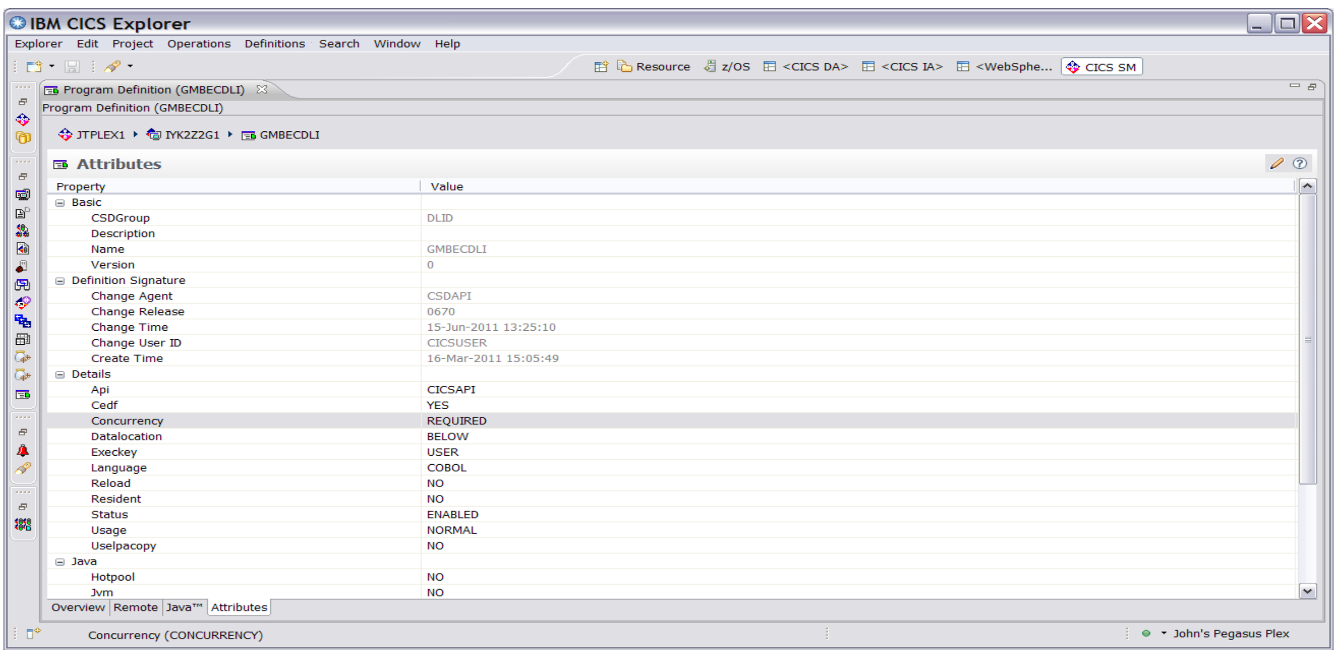

Figure 1. CICS Explorer™ Program definition view showing the CONCURRENCY attribute

The CONCURRENCY(REQUIRED) option allows the user to define that the program must start on an open TCB independently of defining what APIs it uses.

- If the program uses only CICS supported APIs (including access to external resource managers such as DB2, IMS, and WebSphere MQ®), it should be defined with program attribute API(CICSAPI). In this case CICS always uses an L8 open TCB, irrespective of the execution key of the program, because CICS commands do not rely on the key of the TCB.

- If the program uses other non-CICS APIs, it must be defined with program attribute API(OPENAPI). In this case CICS uses an L9 TCB or an L8 TCB depending on the execution key of the program. This allows the non-CICS APIs to operate correctly. This OPENAPI behavior is the same as in previous releases.

Existing threadsafe applications, which have taken advantage of the performance gains of being able to run on the same TCB as the call to an external resource manager such as DB2 or WebSphere MQ by being defined as CONCURRENCY(THREADSAFE) API(CICSAPI), may be able to gain further throughput advantages by being defined as CONCURRENCY(REQUIRED) API(CICSAPI). Throughput gains are achieved when an application can run for longer periods of time on an open TCB.

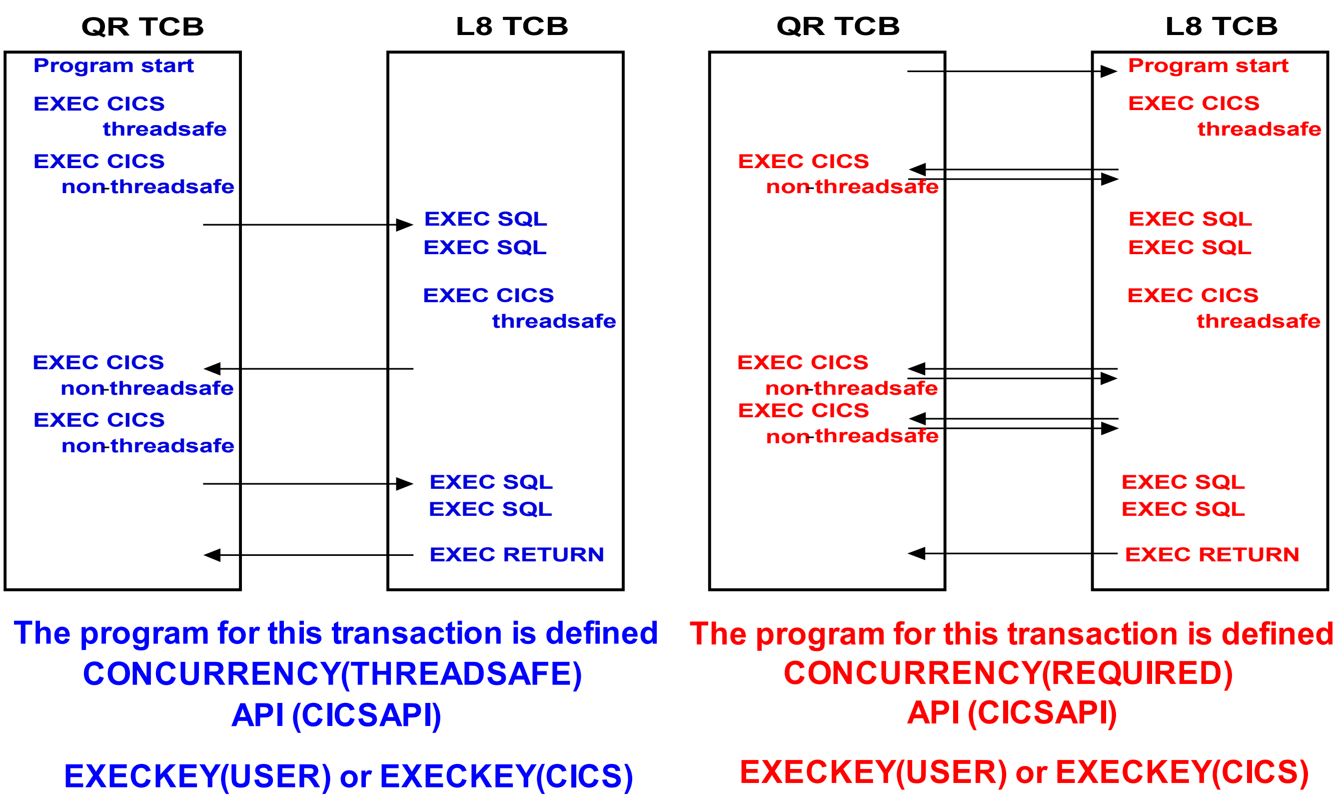

Figure 2. Comparison of CONCURRENCY(THREADSAFE) and CONCURRENCY(REQUIRED)

However, not all applications will be suitable. For example, a threadsafe application that issues a large number of EXEC SQL requests and then issues a large number of EXEC CICS commands that are not threadsafe, is best left as CONCURRENCY(THREADSAFE). Defining the application as CONCURRENCY(REQUIRED) would mean two TCB switches for each non-threadsafe CICS command, because control always returns to the application on the open TCB as shown in Figure 2.

This situation demonstrates the importance of knowing what the application does. To help you find out, tools such as CICS Interdependency Analyzer for z/OS (CICS IA) provide the ability to discover application execution paths. In particular its command flow feature shows the order in which CICS, IMS, WebSphere MQ, and DB2 commands run, and what TCB each command ran on. Other tools such as CICS Performance Analyzer for z/OS (CICS PA) analyze CICS performance SMF data and show, for example, how much CPU has been consumed on which TCBs, and how many TCB switches have occurred. Tools such as CICS IA and CICS PA are invaluable aids to have in your toolbox when embarking upon a threadsafe project.

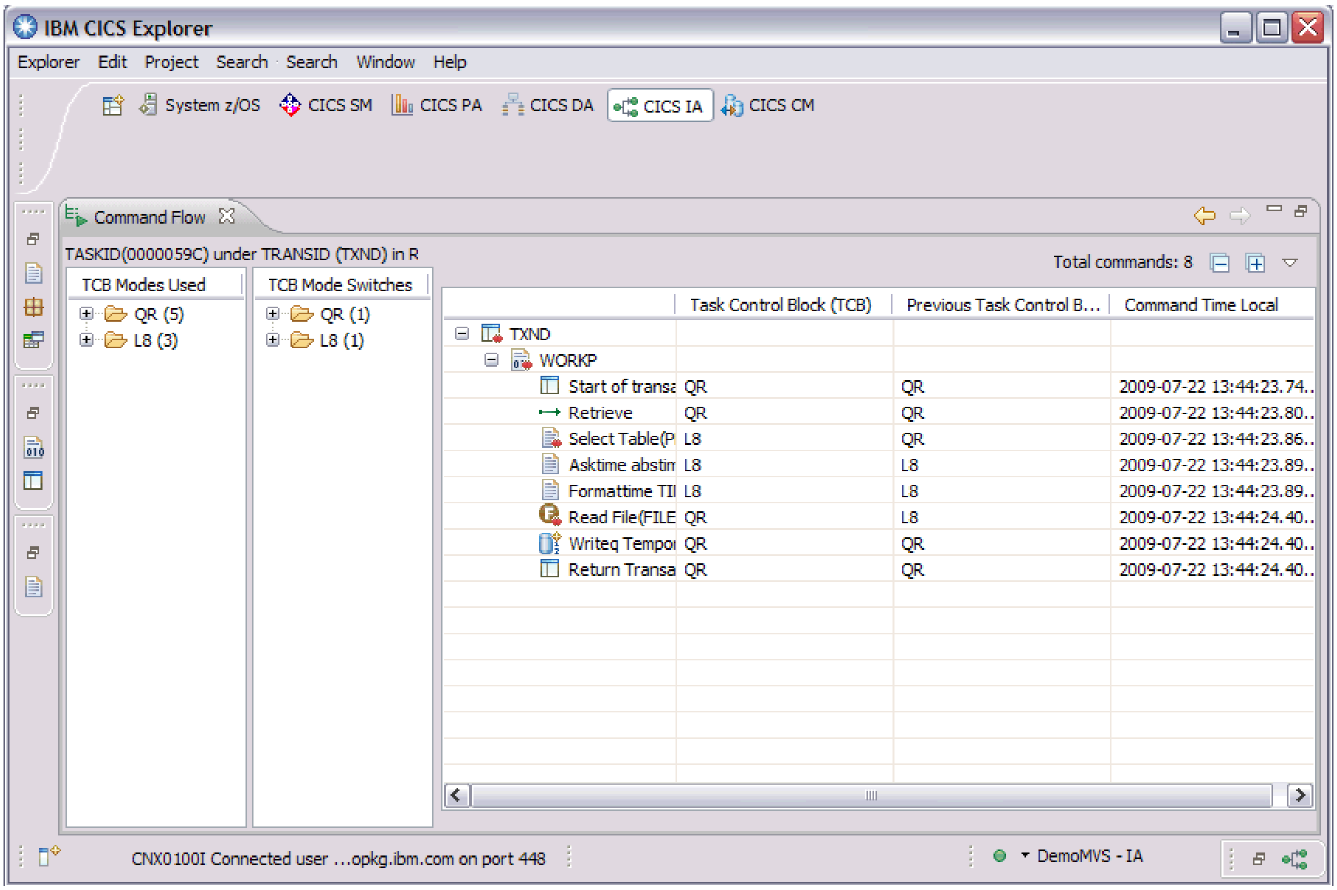

Figure 3: CICS IA plug-in to CICS Explorer showing command flow analysis regarding TCB switches

Threadsafe mirrors

The CICS-supplied mirror program DFHMIRS, which is used by all mirror transactions, is now defined as threadsafe. In addition the IPIC transformers have been made threadsafe. For IPIC connections only, CICS runs the mirror program on an L8 open TCB whenever possible. For threadsafe applications that function ship commands to other CICS regions using IPIC, the resulting reduction in TCB switching improves the performance of the application compared to other intercommunication methods. To gain the performance improvement for remote files, you must specify the system initialization parameter FCQRONLY=NO in the file-owning region (FOR).

For remote File Control or temporary storage requests shipped over IPIC connections, CICS will no longer force a switch to the QR TCB in the application-owning region (AOR) if it is running currently on an open TCB. The requests will be shipped running on the open TCB.

In the FOR or queue-owning region (QOR), the mirror decides when to switch to an open TCB. It does so for the first File Control or temporary storage request received over an IPIC connection. The idea is for long-running mirrors to keep the mirror running on an open TCB.

A new option MIRRORLIFE has been added to the IPCONN attributes for function-shipped file control and temporary storage requests using an IPIC connection. MIRRORLIFE improves efficiency and provides performance benefits by specifying the lifetime of mirror tasks and the amount of time a session is held.

Threadsafe CICS-DBCTL interface

CICS provides a CICS-IMS Database control (CICS-DBCTL) interface to support CALL DLI and EXEC DLI requests that are issued by applications running in a CICS region. In CICS TS 4.2 the CICS-DBCTL interface has been enhanced to exploit OTE and CICS can run the CICS-DBCTL task-related user exit (TRUE) on an L8 open TCB.

OTE is supported from IMS Version 12 with PTFs for APAR PM31420 applied. IMS indicates to CICS during the connection process that the OTE is supported and consequently CICS defines the CICS-DBCTL TRUE as an OPENAPI TRUE. For IMS Version 11 and earlier, OTE is not supported and CICS runs the CICS-DBCTL TRUE on the QR TCB and the IMS code switches to an IMS thread TCB.

Running an application on an open TCB improves throughput and performance by reducing the use of the QR TCB. Threadsafe CICS applications that run on an L8 open TCB and issue CALL DLI or EXEC DLI commands can avoid two TCB switches for each call to IMS.

- For a non-threadsafe application there is no reduction in the amount of switching. Instead of switching from the QR TCB to an IMS thread TCB and back again for each IMS request, the application switches from QR to L8 and back again.
- For a threadsafe application, if it is running on the QR TCB it switches to L8 and then stays on L8 when control is returned to the application.
- For a threadsafe application that is already running on an L8 TCB, or for a CONCURRENCY(REQUIRED) application that is running on an L8 TCB, no TCB switching occurs for the IMS request.

Threadsafe SYNCPOINT and other commands

CICS commands that have been made threadsafe in CICS TS 4.2 include Named Counter Sever commands, QUERY SECURITY, SIGNON, SIGNOFF, VERIFY PASSWORD, CHANGE PASSWORD, EXTRACT TCPIP and EXTRACT CERTIFICATE, along with a number of new SPI commands. Most significant however is the SYNCPOINT command.

The CICS Recovery Manager domain now processes a SYNCPOINT command on an open TCB wherever possible to minimize TCB switching. Syncpoint processing can take place on an open TCB for all resource types declared as threadsafe that were accessed in the unit of work. If resource types not declared as threadsafe were accessed in the unit of work, the Recovery Manager switches to the QR TCB for those resource types. Prior to CICS TS 4.2, CICS would switch to the QR TCB prior to the end of task sync point. In CICS TS 4.2 the application remains on an open TCB, if it is running on one, until end of task sync point has been called. Afterwards, CICS switches to QR for the task detach logic.

Prior to CICS TS 4.2 a threadsafe application running on an open TCB that had, for example, updated DB2 and WebSphere MQ and then issued a sync point would require 9 TCB switches:

- A switch to QR would be made at the start of the sync point.
- Switches to L8 and back to QR would occur when calling DB2 for PREPARE
- Switches to L8 and back to QR would occur when calling WebSphere MQ for PREPARE
- Switches to L8 and back to QR would occur when calling DB2 for COMMIT
- Switches to L8 and back to QR would occur when calling WebSphere MQ for COMMIT

In CICS TS 4.2, if a transaction is terminal driven one TCB switch to QR will occur. For a non-terminal-driven transaction (and assuming no other non-threadsafe resources had been touched) then no TCB switches occur.

Introduction to 64-bit exploitation

CICS TS 4.2 contains major changes to provide a CICS domain architecture environment that exploits the underlying z/Architecture® for 64-bit addressing, and to provide the infrastructure for the future. This has allowed exploitation by CICS services in CICS TS 4.2, and lays the foundations for future CICS applications to be able to use and exploit 64-bit addressing mode. The exploitation of the 64-bit addressing that is provided by the z/Architecture enables CICS to remove some of the previous limitations that affect scalability and availability by delivering large address spaces.

CICS can use z/OS 64-bit virtual storage to increase capacity by supporting a larger number of concurrent users and concurrent transactions. In addition, CICS can keep up with the virtual storage demands of increased workload of existing applications and the larger memory requirements of new applications and new technologies.

64-bit domains

CICS domains can use stack storage, domain anchor storage and can allocate domain control blocks in virtual storage above the bar. Kernel, Monitoring, Storage Manager, Lock Manager, Trace, Message and Temporary Storage are all CICS domains that now run AMODE 64 and keep their data above the bar.

Storage

The z/OS MEMLIMIT parameter limits the amount of 64-bit (above-the-bar) storage that the CICS address space can use. This storage includes the CICS dynamic storage areas above the bar (collectively called the GDSA) and MVS storage in the CICS region outside of the GDSA.

A CICS region requires at least 4 GB of 64-bit storage. You cannot start a CICS region with a MEMLIMIT value that is lower than 4 GB. If you attempt to do so CICS issues a message and terminates.

Note: CICS does not try to obtain the MEMLIMIT amount of storage when initializing; 64-bit storage is obtained as required.

CICS Storage Manager domain has been greatly enhanced to manage 64-bit storage and to provide additional statistical information about 64-bit storage consumption. In addition, for 31-bit storage, the minimum and default EDSALIM values have changed to 48 MB to ensure that there is sufficient storage for CICS initialization.

CICS services that exploit 64-bit storage

CICS temporary storage is one of the major exploiters of 64-bit storage in CICS TS 4.2. TS main temporary storage queues can now use 64-bit storage. CICS provides new facilities so that you can check the storage use of main temporary storage queues and limit that storage use. Auxiliary temporary storage queues and shared temporary storage queues continue to use 31-bit storage.

Main temporary storage is in 64-bit storage rather than 31-bit (above-the-line) storage, depending on the version of the z/OS operating system and whether the CICS region operates with transaction isolation. If your CICS applications use large amounts of main temporary storage, the move to 64-bit storage can increase the available storage elsewhere in your CICS regions. Additional new capability is provided to be able to clean up unwanted queues after a specified time interval.

CICS trace domain exploits 64-bit storage (depending on the version of the z/OS operating system being used and whether the CICS region operates with transaction isolation), by allocating the CICS internal trace table above the bar. This provides virtual storage constraint relief for 31-bit storage and allows for very much larger trace tables to aid problem determination. Trace control blocks and transaction dump trace tables also move above the bar, as do message tables used by the Message domain.

For CICS Java applications, all JVMs now run in AMODE 64 instead of AMODE 31, increasing the capacity for running more JVMs in a CICS region. JVM servers and pooled JVMs use 64-bit storage, significantly reducing the storage constraints in a CICS region for running Java applications. You can therefore reduce the number of CICS regions that run Java to simplify system management and reduce infrastructure costs. You can also use System z Application Assist Processors (zAAPs) to run eligible Java workloads.

Conclusion

In conclusion, the scalability enhancements provided by CICS Transaction Server for z/OS Version 4.2 provide, via the OTE enhancements, the ability for more workload to exploit the power of the mainframe. The 64-bit enhancements provide the ability to scale vertically, and therefore do more work in a single CICS region, as well as providing a foundation for even greater capacity in the future.

Further reading

1. Scalability enhancements in CICS TS 4.2: http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.whatsnew.doc/themes/theme5.html

2. Threadsafety learning path: http://publib.boulder.ibm.com/infocenter/cicsts/v4r2/topic/com.ibm.cics.ts.doc/lpaths/threadsafe/overview.html

3. *Threadsafe Considerations for CICS* Redbooks® publication: http://www.redbooks.ibm.com/abstracts/sg246351.html