
Rule CIC103: Maximum Task Class was reached too often

Finding: CPExpert has detected that the CICS region reached the maximum task class specification (CMXT value) too often.

Impact: This finding has a MEDIUM IMPACT on the performance of the CICS region. This rule does not apply with CICS Version 4.1.

Logic flow: This is a basic finding, based upon an analysis of the daily CICS statistics.

Discussion: The Maximum Task Class (CMXT) operand in the System Initialization Table (SIT) limits the total number of concurrent **active** tasks associated with particular task classes. Up to 10 unique task classes can be defined, and a maximum number of active tasks is specified for each class.

Tasks are assigned to classes using the TCLASS operand in the Program Control Table (PCT). For example, CMXT might be defined as "CMXT=(10,4,4,20)" in the SIT. Four task classes are thus defined. Class 1 has a maximum of 10 active tasks, Class 2 and Class 3 have a maximum of 4 active tasks, and Class 4 has a maximum of 20 active tasks.

Note that after CICS Version 1.7, a Dispatch Control Area (DCA) is built for any task received from VTAM regardless of whether the MXT or CMXT ceiling has been reached. The CMXT values do **not** limit the number of tasks which are ATTACHED. Rather, the CMXT values are applied to the task classes when a task is to be considered for dispatching.

After the ATTACH has been processed, CICS considers whether the task should be dispatched. A Task Control Area (TCA) is created at dispatch time if the region is below MXT or if the task class is below CMXT. It is at this point that the CMXT value is used.

Tasks assigned to a task class are selected for dispatching only if fewer than CMXT tasks are already active for the class. If the region is at the MXT or CMXT limit at dispatch time, the TCA is not created.

There are several situation in which the CMXT value may be used to limit the number of active tasks in a class.

- The CMXT specifications can be used to control tasks that may be heavy resource users. This control can limit the amount of resources required to support the heavy resource users (either because management wishes to restrict the amount of resources allowed to

CICS or because management wishes to restrict the resources devoted to specific tasks).

- The CMXT specifications can be used to control low priority tasks. This control can allow processor resources to be used by more important tasks.
- The CMXT operand can be used to ensure that one type of transaction does not monopolize CICS.
- CMXT can be used to control the use of virtual storage by the Dynamic Storage Area (DSA). When the number of dispatchable tasks reaches the CMXT value, tasks on the active chain are not dispatched. Consequently, the tasks generally have little associated storage, or their storage is likely to be paged out of central storage. The CMXT value can be particularly useful if there are a number of long-running tasks and the MXT value cannot be successfully used to limit the amount of virtual storage used by the tasks.

The CMXT operand normally should not be used to serialize tasks. Rather, ENQ should be used for this purpose.

There are a number of disadvantages to limiting the maximum number of active tasks.

- CICS performance can be unnecessarily degraded if the CMXT value is used to restrict the number of concurrent active tasks in a particular class, and if the system is capable of handling more tasks.
- CMXT is not normally suited for conversational transactions, because users can be locked out for a lengthy period if the CMXT value is reached.
- Tasks assigned to a particular class are selected from the dispatchable chain based upon priority. Active tasks may not be dispatched because they are waiting for some event, such as waiting on I/O completion. If the CMXT value is too low, it is possible that no tasks are dispatched even though low-priority dispatchable tasks may be on the active chain. (Recall that the AMXT operand controls how far down the dispatch chain CICS task control searches looking for a dispatchable task. It **does not** control the number of dispatched tasks, but controls the number of **potentially** dispatchable tasks.)
- A temporary lockout can occur if the CMXT value is set too low in an environment where tasks are dependent upon the completion of events processed by other tasks. For example, one task might attach another task and then wait for the completion of an event processed

by the attached task. If the attached task is assigned to a task class and the task class is at its maximum, considerable delay may result.

CPEXpert produces Rule CIC103 if the CICS statistics reported that the maximum active tasks value for any class was reached more than the CMAXn guidance value for the class, and if the maximum number of concurrently attached tasks for the task class was greater than the CMXT specification in the SIT for the task class.

Suggestion: CPEXpert suggests that you review the CMXT value in the SIT. Under most circumstances, the CMXT ceiling should be reached infrequently. If the ceiling is frequently reached, you may wish to increase the ceiling.

Alternatively, reaching the CMXT ceiling may simply be an indication of a performance problem elsewhere in CICS. If CICS performance is poor, tasks will not quickly flow through the system and the CMXT ceiling may be reached. Tuning CICS areas may cause tasks to move through the system quicker and the CMXT ceiling may not restrict the number of tasks dispatched in the class.

Alternatively, you may wish to direct tuning efforts at the specific tasks assigned to the class. Perhaps the resource demands of the tasks can be reduced.

Alternatively, the resources used by the tasks can be a constraint. For example, DASD constraints may impede the flow of tasks assigned to the class. Perhaps a improvement in the DASD configuration, reorganizing file placement on volumes, or other actions would reduce the DASD constraint and allow the tasks to more quickly flow through the system.

If the CMXT value is appropriate for your management objectives and CICS performance cannot be improved, then you may wish to change the CMAXn guidance value for the class to prevent spurious production of this rule.

NOTE: The significance of this finding depends upon whether the finding is based upon analyzing daily information or based upon analyzing historical information.

- If this finding is based upon an analysis of daily information, the finding may be applicable only to the performance of CICS for this day. Unless you feel that the analysis is generally applicable (or unless the workload processed on this day is particularly critical), please wait until CPEXpert performs an analysis of historical information before taking action.
- If this finding is based upon an analysis of historical data covering a

prolonged period, the finding is more definite than a tentative finding based upon analysis of only a single day's data.

Reference: *CICS/OS/VS Version 1.7 Performance Guide*: pages 48 and 199.
CICS/MVS Version 2.1.2 Performance Guide: pages 242 and 377.
CICS/ESA Version 3.1.1 Performance Guide: pages 123 and 298.
CICS/ESA Version 3.2.1 Performance Guide: pages 194 and 337.
CICS/ESA Version 3.3.1 Performance Guide: pages 204 and 355.
CICS/ESA Version 4.1.1: not applicable.
CICS Transaction Server for OS/390: not applicable
CICS/TS for z/OS: not applicable

|

|