
Rule WLM015: Execution velocity goal specified for TSO Period 1 or Period 2

Finding: An execution velocity goal was specified for TSO Period 1 or Period 2.

Impact: This finding should be viewed as generally having a LOW IMPACT or MEDIUM IMPACT on the performance of the service class periods involved. The level of impact depends on the value specified for the execution velocity.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Installations may specify an *execution velocity goal* for a service class period. An execution velocity is a measure of how fast work should run when the work is ready to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage¹. The purpose of specifying an execution velocity goal is to allow installations to specify how important it is for work to have access to a CPU, when the work has no time-related measure (that is, a response requirement is not associated with the work).

The execution velocity is computed based on samples collected at periodic sampling intervals² by the Workload Manager (SRM). The SRM sampling code interrogates address space control blocks (TCBs, SRBs, OUCBs, and OUXBs) to determine the state of each address space assigned to a service class. Sampling counts associated with the service class are updated based upon the state³ of the address spaces.

Please refer to Rule WLM104 for additional discussion of an execution velocity goal.

A service class representing TSO Period 1 or TSO Period 2 transactions normally has a *response goal* defined. This is because these TSO transactions normally are quite interactive, and response time is an appropriate measure of performance. These transactions may be contrasted with TSO Period 3, for example. TSO Period 3 typically contains non-interactive transactions, even though the transactions were

¹Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

²With MVS/ESA SP5.1, the sampling interval is 250 milliseconds. The state of each TCB or SRB associated with an address space is sampled every 250 milliseconds, beginning from address space initiation.

³Note that an address space can be in multiple states (for example, a CICS region might be using multiple processors concurrently, while some CICS tasks were also waiting on some function). Thus, the sample counts can total more than 100% of the sample intervals for any address space.

submitted under TSO. Since TSO Period 3 normally represents non-interactive transactions, a response goal is inappropriate for TSO Period 3. Instead of a response goal, either an execution velocity goal or a discretionary goal is usually specified for TSO Period 3.

When an execution velocity goal is specified for TSO Period 1 or TSO Period 2, the Workload Manager determines whether the TSO transactions *receive* CPU service when they *require* CPU service. Thus, the goal against which performance is being evaluated is **resource oriented** rather than **response time oriented**.

Users, of course, are not as concerned with resource performance as they are concerned with response time. Consequently, service agreements normally are written in terms of response time. Response time normally should be the goal against which the Workload Manager is measuring performance for TSO Period 1 or Period 2.

There are several system-related disadvantages to specifying a velocity goal for TSO Period 1 or Period 2:

- The SRM controls swap protect time on an **individual address space** basis for service classes with an execution velocity goal, rather than controlling swap protect time on a **service class** basis as is done for response goals. Controlling swap protect time on an individual address space basis adds unnecessary overhead.
- If the TSO transactions take awhile to complete (and an execution velocity goal has been specified), the SRM will examine TSO address spaces to decide what expanded storage access to give to their demand paging, VIO paging, and hiperspace paging.
- The SRM might implement working set management algorithms to monitor the TSO address space if an execution velocity goal is specified for the TSO service class. This overhead is not required if a response goal is assigned to the TSO service class.

CPExpert produces Rule WLM015 when an execution velocity goal was specified for TSO Period 1 or Period 2.

The following example illustrates the output from Rule WLM015:

RULE WLM015: EXECUTION VELOCITY GOAL SPECIFIED FOR TSO PERIOD 1 OR 2.

CPEXpert noticed that an execution velocity performance goal was specified for a service class (Period 1 or Period 2) that had "TSO" as a part of its description. Under most circumstances, you should specify a response goal for TSO Period 1 or Period 2. Not only is it more efficient from a Workload Manager view to specify a response goal for TSO, but users tend to relate well to response time (rather than execution velocity). There are a few situations in which you may wish to specify an execution velocity goal for TSO Period 1 or Period 2. These are described in the WLM Component User Manual. Unless these unusual situations apply in your environment, CPEXpert suggests that you change the performance goal from an execution velocity goal to a response goal for the TSO Service Class (Period 1).

Suggestion: CPEXpert suggests that you review the execution velocity goal specified for the service class periods listed with Rule WLM015. CPEXpert suggests that you consider changing the goals to specify a response goal for the TSO service class periods identified by this rule.

There are some situations in which you may wish to specify an execution velocity goal for a TSO Period 1 or Period 2 service class. **If these situations apply, please ignore the above suggestion.**

- You may have established a service class for TSO transactions, but expect to have very few transactions execute in the service class. If few transactions execute in a service class, the Workload Manager would not have any "history" of response time from which to compute average response time or meaningful percentile response time. If the Workload Manager cannot compute the average or percentile response time, insufficient information would be available for the Workload Manager to make its decisions. In such a situation, you may have specified an execution velocity simply to ensure that the TSO transactions received some level of service.
- You may have an unusual situation in which you have established a service class for TSO transactions which typically take a long time to run. Some organizations use TSO to perform functions that normally might be considered batch jobs. If you have such a situation, you may have specified a TSO service class with an execution velocity for either Period 1 or Period 2. Since a response goal would be inappropriate, the execution velocity goal may be an appropriate specification.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5): Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R1): Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R2): Chapter 8: Defining Service Classes and Performance Goals

OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Effective Use of MVS Workload Manager Controls", Ed Berkel and Peter Enrico, IBM Corporation, CMG Transactions, Winter 1995, the Computer Measurement Group.