WLM: Managed Initiators

OS/390 R4

Author: Mike Cox. Document Date: October 7, 1997 - Updated 10/15/02 by Jim McCoy

Abstract

This document describes the Workload Manager Initiator Management function introduced by the Workload Manager component of OS/390 R4 and exploited by the JES2 component in OS/390 R4 for batch jobs.

Reader Responsibilities: This document is a work in progress. The reader is encouraged to make constructive comments on content, style, organization, areas needing additional explanation, etc.. Please send your comments to Jim McCoy at the following address:

jemccoy@us.ibm.com

If you wish to send a fax, the number is:301-240-2590 (or tie-line 8/372-8136).

Trademarks

Special Notices

Acknowledgments

This document, to say nothing of the function, is due to the efforts of many people including:

Jeff Aman	WLM Development, S/390 Division
John Arwe	WLM Development, S/390 Division
David Booz	WLM Development, S390 Division
Greg Dritschler	WLM Development, S/390 Division
John Kinn	JES2 Development, S/390 Division
Laurie Letersky	SDSF Development, S/390 Division
Tom Wasik	JES2 Development, S/390 Division
Peter Yocom	WLM Development, S/390 Division

Table of Contents

Overview and Background

This chapter is intended to provide some insight into the requirement for WLM initiator management and the approach taken by the Workload Manager component and the JES components in managing batch work flow from job submission to execution completion. There is also a discussion on JES job class usage and job delays prior to the execution phase that are pertinent to WLM management of initiators.

Overview

Prior to OS/390 R4 and WLM managed initiators, getting the job started and getting the job to run were essentially two independent, and occasionally decoupled, processes. Assuming the installation has the perfect 'IEAICSxx and IEAIPSxx' for their configuration, strong job naming conventions and stringent enforcement of proper job class assignment is required to have consistency between job selection and job execution.

Despite all of the tools to manage batch, certain aspects of batch workflow have remained the same since the invention of the Job Entry Subsystems (JES2 or JES3).

> *Batch jobs are selected for execution based on the availability of an initiator requesting a job from a specific job class queue.*

JES's simple approach for selecting work ignores installation requirements for:

- Selecting 'the most important work'.
- Selecting work when 'execution resources are available'.
- Initiating the work on the 'correct MVS image.'
- Not selecting work when there is no capacity to run additional work.
- Adjusting to changes in the backlog of batch jobs awaiting execution.

A great amount of time and effort is expended answering the following questions because JES managed initiators cannot make intelligent choices.

- How many initiators are needed on each MVS?
- How many initiators of each type (i.e., what is the selection criteria) on each MVS?
- How does either of the above change from shift to shift, during the weekend, during a closing period, following a machine upgrade, etc.

- How is important work kept from being impeded by less important work, but still get all the work done?
- How does the job class and job priority specification correspond to performance groups, domains and allotted MPL levels across the MVS images in the JESplex.

WLM managed initiators provide the capability of integrating work selection, or batch initiation, into installation defined goals for producing completed units of work. Furthermore, the management of the batch job queue adjusts dynamically in response to work availability and processor resources.

In a Nutshell

Workload Manager (WLM) has provided a mechanism for classification of work and the allocation of system resources (e.g., CPU, memory) to units of work during the execution phase. WLM has consistently enforced installation defined rules and goals across all MVS images in the sysplex. However, the initiation of batch work has not been within the control of WLM. WLM has previously worked in reaction mode as JES placed jobs into execution independent of current conditions. Beginning in OS/390 R4, WLM has the capability of controlling the rate at which queued jobs are initiated. Moreover, the WLM will dynamically change the number of WLM initiators and/or their work selection criteria in an attempt to meet installation defined goals.

Both WLM and JES required substantial changes for WLM management of batch job initiation. JES has been extended to classify batch jobs following converter processing and provide job backlog information to WLM, by service class, on each image in the JESplex. WLM acquires the job backlog information and uses it in determining whether or not installation goals are being met. Furthermore, WLM starts, stops, or changes job selection criteria for WLM managed initiators based upon availability of work, goal attainment and processor resource availability within the sysplex. A simple concept - the implementation is just a matter of details.

However, the key points are:

- WLM has knowledge of the available job backlog by service class for each image in the JESplex.
- WLM has total control over its initiators, including the number of initiators, the type of work the initiators select, and when the initiators select it.
- The installation has control over the categories of batch work to be executed in WLM managed initiators.
- WLM managed initiators and JES managed initiators can coexist in the same JES environment.
- Started procedures (or tasks), started jobs and TSO logons are not affected by WLM management of initiators. These types of work will continue to be managed by JES as they have in the past.

Background

Before discussing WLM management of initiators, lets take a quick review of the ways JES job classes are used and how batch jobs are delayed from being selected for execution. Today, these ‘externals’ are critical to batch job management and will continue to be used in the new environment.

Job Classes

JES job classes are used for many different purposes; some classifications are well founded and some are political arrangements. Nonetheless, most installations have established standards for assignment of a batch job to one of 36 (JES2) or 255 (JES3) categories known as job classes. These include:

- Workload importance

  Batch work normally falls into one of three categories:

  - Hot, do it right now, or
  - Scheduled production, or
  - Unscheduled, do as capacity permits.

  Within each of these categories are minor specifications: Production, Development, Test, etc..

- Political compromises

  A job class and some number of initiators are dedicated for use by a specified group within the installation.

- Resource Control

  Classes established for jobs that require large numbers of tape drives, large amounts of work pack space, large amounts of processor storage, etc..

- Resource Scheduling

  Classes established for jobs which should only run at certain times of the day, or need access to resources that are not available at all times throughout the JESplex.

- Attribute assignment

  Jobs are assigned attributes by being placed in a specific job class. These attributes include:

  - Journaling
  - Restart options
  - MVS command authority (JES2)
  - Time Limits (JES2)
  - Bypass Label Processing (JES2)
  - Spool partition assignment (JES3)

These categories are not necessarily mutually exclusive nor are they exhaustive. However, it is important to understand the use of job classes when considering the implementation of WLM managed initiators.

Job Delays Prior to Execution

Management of the batch job queue is essentially an exercise in getting the correct jobs started at the correct time on the proper system. There is no mechanism to tell JES to ‘initiate batch job number 2345, now’. Initiating a set of selected jobs or types of jobs is accomplished by keeping all other jobs from being selected. The operational techniques for preventing jobs from being selected are accomplished using a combination of the JES's initiator scheduling capabilities and installation unique procedures.

There are many reasons why a job submitted at certain time should not be sent into execution immediately. Otherwise, installations could dispense with batch jobs and just use started tasks. (On the other hand, there are also occasions when it is desirable to force batch jobs to start immediately.) There are several methods of preventing batch jobs from being selected for execution. The following list categorizes many reasons preventing a job from being selected. JES differences are noted where pertinent.

- User specified delay
  - TYPRUN=HOLD or TYPRUN=JCLHOLD (JES2) specified on the job card.

    The submittor of the job is giving explicit instructions to JES to prevent the job from executing until the operator (or user) releases the job.
- Operational delay
  - Non-selectable job class

    The job is associated with a job class for which there are no initiators selecting work. The job will not be eligible for processing until the job's class is re-set to a class for which work is being processed.
  - Initiators unavailable

    The job is associated with a job class which is normally a scheduled job class. However, no initiators are started that allow selection of work in the class.
  - System unavailable

    The job has an affinity to a system that is currently unavailable for scheduling work.
  - Job class held (JES2)

    The job is associated with a job class which is held. Initiators cannot select jobs from this class.
  - Job class off (JES3)

    The job is associated with a job class which is ‘OFF’. Initiators having the designated off class in their selection criteria are prevented from selecting work in an off class.
  - Job priority held (JES3)

    The job is assigned a job priority which is held. Initiators having the designated class in their selection criteria are prevented from selecting work in a held priority.

  - Job queue hold

    All jobs in the job queue are prevented from being selected for execution.

  - Spool hold

    The job is associated with a spool volume for which activity is not allowed.

- Resource delays

  - Resource Affinity Scheduling

    Resource affinity scheduling is a new mechanism in OS/390 R4 allowing a job to be delayed for execution until the required resources are available on some image in the JESplex.

  - Setup delays (JES3)

    The installation utilizes pre-execution setup for data sets and/or devices. When the job requires incompatible usage of a data set or device in use by another job the job is queued, waiting for the resource to become available.

- JES processing delays

  - Reader and Converter time

    This is the elapsed time that a job is held in reader (input service processing) and converter processing.

- JES scheduling delays

  - Duplicate job name

    JES3 and JES2 (optionally) will prevent two or more jobs with the same name from executing at the same time.

  - Dependent job control (JES3)

    DJC is a mechanism to control the execution of a collection of jobs in a designated order. Jobs lower in the hierarchy are prevented from selection until ‘released’ by the completion of jobs higher in the hierarchy.

  - Job class limits (JES3)

    The installation has placed limits on the number of active jobs of a given class based on the total number of jobs of the specified class active on a particular processor, in total or relative to the number of jobs of other classes active in the JES3 complex

- Execution delays

  - Waiting for selection

    The job is available for initiator selection and is basically waiting until an initiator 'finds it and selects it.' At this point, higher priority work could arrive and be selected ahead of this job.

Prior to OS/390 R4, all of the elapsed time from end of converter phase to job select was considered to be queue time. Beginning with OS/390-JES2 R4, JES queue time is categorized into four distinct buckets. They are:

Category 1: JES processing delays.
Category 2: Operational delays and JES scheduling delays.
Category 3: Resource scheduling delays.
Category 4: Execution delays.

Note: User delay caused by specifying TYPRUN=HOLD or JCLHOLD is not counted in any category. The time period from when a job is submitted to when it is allowed to progress is not counted as queue time. Accumulation of time in the various categories of JES queue time categorization only occurs when the job is released from the hold condition.

Installations have long been concerned with delays leading up to execution. Capturing initiator selection delays into these categories allows the installation to address delays within their control. Moreover, WLM must now be concerned with the delays in Category 4 as it is to required to manage job queue time to meet installation objectives for throughput and response time. To this end, the SMF30 record has new fields that capture this information for installation analysis.

Which Bucket?

Consider the situation where a job, JOBONE, is submitted into a JES complex of 3 members at 10:00:00 AM. The job has a scheduling environment requirement associated with it as well. Further suppose it takes JES 5 seconds to process the job through input service and converter processing. At 10:00:05, JES is attempting to schedule the job into execution. However, the situation at 10:00:05 is:

- SY1 **-** there are no initiators selecting work from the job class associated with the JOBONE. However, the scheduling environment is available on SY1.

- SY2 **-** there are initiators available attempting to select jobs in JOBONE's job class. However, the job is in resource hold, due to the unavailability of the job's requested scheduling environment.

- SY3 **-** the job is waiting for an initiator as all the initiators eligible to select work in JOBONE's class are occupied. The scheduling environment is available on SY3..

Now suppose the job's status remains constant on all images until 11:00:00 at which time any one of the following events could occur:

- **Case i -**

  On SY1, an initiator is started at 11:00:00 which is eligible to select the jobs from the job class to which JOBONE is assigned and the job is immediately selected for execution.

- **Case ii -**

  On SY2, the scheduling environment is made available at 11:00:00, an initiator is available and the job is selected for execution.

- **Case iii -**

  On SY3, at 11:00:00 the scheduling environment is available , an initiator becomes free and the job is selected for execution.

The question is:

> *How much time is accumulated in the previously described delay buckets for JOBONE should case i, case ii or case iii occur? Does it matter which case occurs?*

All should agree that 5 seconds is the value for Category 1 delay. How much of the 59 minutes and 55 seconds should go into categories 2, 3, and 4 for each of cases listed above? In fact, it doesn't matter which case occurs, all 59 minutes and 55 seconds will be placed in Category 4.

Why is all the time assigned to 'execution delay' you may ask? The rationale for this assignment of time into the execution delay category is:

> *In the situation where a job is delayed for different reasons on multiple JES instances concurrently, the job delay will be charged to the least restrictive cause of delay.*

The hierarchy for delays is: Category 2 (most restrictive), Category 3, Category 4 (least restrictive). In the above example, on at least one MVS instance, the job was only waiting for an initiator to select it for the 59 minutes and 55 seconds. Thus, the job delay time is attributed to 'execution delay'.

WLM and JES

This chapter provides an overview of the process whereby JES and WLM assume joint responsibility for placing batch work into its execution phase.

Pre-requisites

WLM

WLM at the OS/390 R4 level or higher must be active in **goal mode in the sysplex**.

JES2

JES2 must be at OS/390-JES2 R4 on all members of the JES2 MAS. Additionally, the $ACTIVATE command must have been executed or OS/390 JES2 R4 initialized with a cold start.

JES Responsibilities

JES continues to own and manage the job queue. In fact, WLM is completely unaware of the batch jobs by name or job number. JES continues to do all the work it has traditionally performed on behalf of jobs prior to the WLM managed initiator implementation. In addition, JES must do the following tasks in support of this new WLM function:

1. Classify all batch work using WLM services.

2. Provide WLM on each member of the JESplex with an accurate inventory of the number of jobs in each service class awaiting initiator selection on each member and across the JESplex as a whole..

3. Keep jobs destined for WLM managed initiators from being selected by JES initiators and vice versa.

The rest is 'just details'... and a substantial amount of code. The following describes areas of change and responsibility between JES and WLM.

Job Classification

JES classifies all batch work following the Converter phase (JES2) using WLM classification services. The service class information assigned by WLM is kept by JES, in a spool resident control block, for the life of the job. The service class is passed to the initiator when the job is selected. Classification processing occurs for all batch jobs regardless of whether they are destined for traditional JES managed initiators or WLM managed initiators.

Job Modification

While a job is awaiting execution, its job class or priority can be changed. Either one of these events causes JES to invoke the WLM classification service to obtain a potentially new service class. JES remembers the original service class definition and the current service class definition. JES passes the current value to the initiator when the job is selected. However, JES records in the SMF26 record the original service class and the service class in use by the job at the completion of execution.

JES Initialization

During initialization, JES registers with WLM, enabling the JES instance to use WLM work classification services. JES determines of the current state of WLM in the sysplex and establishes exit points to detect WLM changes for which it must take action.

JES validates and re-establishes the job queue following a restart. In the case of JES2, this processing occurs on the first member to initialize in the JES2 MAS. Several events could have occurred while JES was absent, including:

- Activation of a new WLM policy
- Modifying the service class of a job while in execution.
- Changing the job classes for which WLM has initiation responsibility.
- Changing of resource scheduling environments.

JES determines whether the WLM policy has changed by comparing the current WLM state token to the WLM state token being used when JES was last active. If this is the first member in the MAS to initialize and the WLM policy is different, then JES re-classifies each batch job waiting for execution.

All jobs in execution must have their current classification updated in JES control blocks (This is necessary for accounting and the current service class for a job is needed for job restart situations.) This processing is performed by JES on each member of the JESplex.

JES also evaluates which jobs are available for initiator selection by WLM. For example, jobs must have all their ‘hold’ conditions removed to be considered eligible for execution. Once the inventory is complete, each JES registers each service class it discovered associated with a batch job destined to or active in a WLM managed initiator. Then each JES instance provides WLM with a list of service classes and the number of jobs in each service class that are available to be selected on this MVS instance and throughout the JESplex.

JES Termination

When JES terminates (normally or otherwise), the collection of service class queues to be initiated by WLM are de-registered from WLM. WLM then knows it no longer has a source of batch work on the effected system.

WLM Policy Changes

While JES is actively processing jobs, a WLM policy can be activated with new or changed service class definitions. Each JES instance detects this event through an ENF signal issued by WLM on the MVS image. Consequently, JES must re-inventory the job queue waiting for initiation. This processing occurs on only one instance of a JES2 MAS. All the jobs must be re-classified by JES and their new service class saved. Each JES instance acquires the service class of all work currently in execution and saves the information as well.

Job selection for JES and WLM managed initiators is suspended on all MVS instances during this reconciliation period. This processing is much the same as that during JES initialization.

Once the job queue is re-classified, each JES instance makes known to WLM a list of service classes with the number of jobs available for execution on that MVS image. Job selection for JES and WLM managed initiators then resumes.

Job Queue Management

JES continues to be responsible for determining when and on which systems a job is eligible for initiator selection. Only JES knows whether or not all of the ‘hold’ conditions, discussed in a previous section, have been satisfied and on which systems they have been satisfied. JES deals with the details of the individual job, whereas WLM is concerned only with the number of jobs in the service class as a whole.

Job Backlog for WLM Initiators

The job’s class is the sole determining factor for whether a job is destined to a JES managed initiator or a WLM managed initiator. Unlike JES initiators, WLM initiators select work by service class. In order for WLM to intelligently manage the backlog of jobs for which it has initiation responsibility, WLM must understand the job backlog at the system level and across the JESplex. Specifically each WLM instance must be provided with a list of each service class for which it is responsible to schedule batch jobs. For each service class, JES must tell WLM:

1. The number of jobs currently available for selection on this system.
2. The number of jobs currently unavailable for selection on this system.
3. The number of jobs currently available for selection somewhere in the JESplex.
4. The number of jobs currently unavailable for selection on any system in the JESplex.

JES determines which service classes must be registered with WLM through discovery. Following the classification of each job, JES must determine if this is the first encounter of this service class. If this is a newly discovered service class and the job class indicates the job should be initiated in a WLM managed initiator, then each JES instance of the complex must register this service class with WLM in order for WLM to be able to select jobs from the service class. From this point in time on, each JES instance in the sysplex provides WLM information regarding the number of jobs in the service class eligible for execution on that MVS instance.

JES also de-registers a service class after an extended period of time in which no jobs are in execution or awaiting execution in the service class. Should a job be submitted that is classified into the de-registered service class, the service class is registered with WLM just as before.

The following example should illustrates these points.

Conceptual representation of WLM's view of available work

SY1

WLM Initiator Service Classes

AA	4	0	1	3
HOT	1	0	1	0
COLD	0	1	0	1
JUNK	3	0	3	0

SY2

WLM Initiator Service Classes

AA	4	0	2	2
HOT	1	0	1	0
COLD	0	1	0	1
JUNK	3	0	0	3

SY3

WLM Initiator Service Classes

AA	4	0	3	1
HOT	1	0	1	0
COLD	0	1	0	1
JUNK	3	0	1	2

JES Complex: SY1, SY2, SY3

Jobs Classes initiated by WLM: A, B

Jobs Classes initiated by JES: C

Jobs Awaiting Execution:

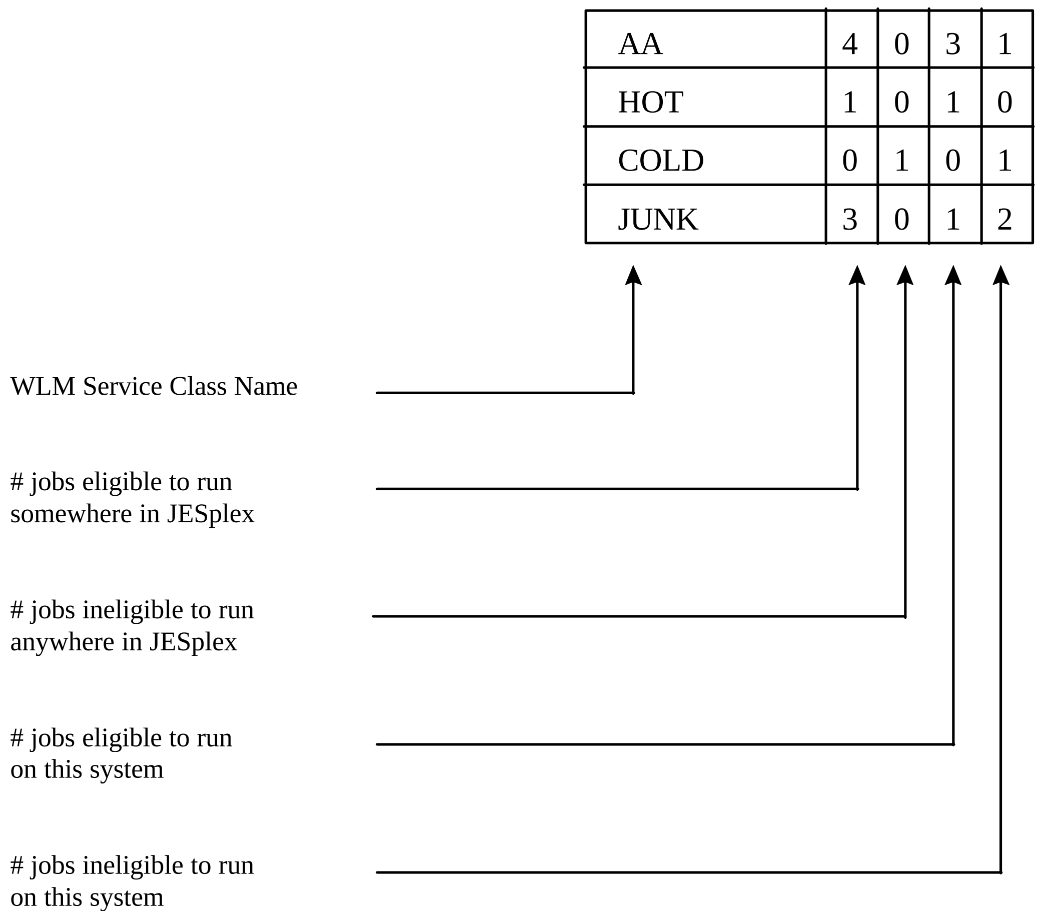
Job Name	Job Number	Job Class	Service Class	System Eligibility		
MCCOXA	J00024	A	AA	SY1	SY2	-
MCCOXB	J00026	A	AA	-	SY2	SY3
MCCOXC	J00027	B	AA	-	-	SY3
MCCOXD	J00123	B	AA	-	-	SY3
MCCOXE	J00230	B	HOT	SY1	SY2	SY3
MCCOXF	J00234	B	COLD	-	-	-
MCCOXG	J00037	A	JUNK	SY1	-	-
MCCOXH	J00043	B	JUNK	SY1	-	SY3
MCCOXI	J00134	B	JUNK	SY1	-	-
MCCOXJ	J00337	C	JUNK	-	SY2	-
MCCOXK	J00154	C	HELLO	-	-	SY3

Notes:

- No jobs in any type of hold condition.

JES provides each WLM instance a view of the work available or its system and a view of work available to other systems in the JESplex. Details of the view are in the following figure:

Job Backlog Reported by JES to WLM by Service Class on SY3



AA	4	0	3	1
HOT	1	0	1	0
COLD	0	1	0	1
JUNK	3	0	1	2

WLM Service Class Name
# jobs eligible to run somewhere in JESplex
# jobs ineligible to run anywhere in JESplex
# jobs eligible to run on this system
# jobs ineligible to run on this system

JES provides each WLM instance a view of the work available for its system and a view of work available to other systems in the JESplex.

Notes for the preceding figures:

- The JESplex has three members: SY1, SY2, and SY3.
- The installation is allowing WLM control the initiation of jobs in class A and B. Jobs in class C are initiated in JES initiators. Once the job is initiated, the execution phase is under the control of WLM regardless of the type of initiator.
- There are 11 jobs on spool waiting for execution. Each job has the obligatory job name, job number, job class, service class, and system eligibility mask. It is not important how this mask is determined for this example, only that jobs are limited to certain processors for execution.
- During the course of accepting these jobs for execution, JES obtained a service class for each of the jobs. Nine jobs (3 - class A, 6 - class B) were determined to be

destined for WLM managed initiators. Two jobs (2 - class C) were determined to be destined for JES managed initiators.

- The nine jobs from class A or B have been assigned to one of four service classes (i.e., AA, HOT, COLD, JUNK). Whenever JES first discovered each of these service classes, it registered the service class with WLM. (Service class registration could have occurred when JES initialized and discovered the jobs already in the job queue. A job could also be submitted in class A or B and be assigned to a service class that JES had not previously encountered. At that point JES would have to register the service class with WLM.) JES only registers a service class with WLM if it has work destined to a WLM managed initiator.

- Each JES instance has registered the same four service classes in this example. JES is obligated to give a JESplex view of service classes containing jobs destined for WLM managed initiators. This allows WLM to derive the JES topology. Remember there could well be more than one JESplex in the sysplex and WLM service classes are sysplex in scope.

- The two jobs from class C have been assigned to one of two service classes (i.e., JUNK, HELLO). As these jobs are queued to JES managed initiators, JES does not register these service classes with WLM. The service classes may be registered with WLM, but this is due to the existence of a job in a job class destined for a WLM managed initiator.

  Job MCCOXJ is assigned the same service class, JUNK, as MCCOXG, MCCOXH, and MCCOXI. Any one of the latter three jobs could have caused the service class JUNK to be registered with WLM. MCCOXJ did not as it is not destined to a WLM managed initiator. It is permitted, though not always desirable, to have JES managed initiators and WLM managed initiators accept the same type of work (by service class).

- Service class HELLO is not registered with WLM, as there is no work destined for WLM managed initiators in this service class.

- On each JES instance, JES must provide an accurate inventory of job back log on this processor and across the JESplex for each registered service class. Each WLM instance is presented a inventory of jobs by service class available to that MVS image. For example:

  - In service class AA, there are a total of four jobs eligible for selection. There are no jobs unavailable for selection on all members of the JESplex; however, no job is eligible for selection on all three MVS images. In service class AA, WLM sees one available job on SY1, two available jobs on SY2, and three available jobs on SY3. Each WLM instance sees 4 jobs available for selection somewhere in the JESplex.

  - In service class HOT, there is one job. The job is eligible for selection on all three MVS images. In service class HOT, WLM sees one available job on SY1, one available job on SY2, and one available job on SY3. Each WLM instance sees 1 job available for selection somewhere in the JESplex.

  - In service class COLD, there is one job. The job is not eligible for selection on any of the three MVS images. In service class COLD, WLM sees zero available jobs on SY1, zero available jobs on SY2, and zero available jobs on SY3. Each WLM instance sees 0 jobs available for selection somewhere in the JESplex and 1 job unavailable for selection anywhere in the JESplex.

  - In service class JUNK, there are three jobs. There are in fact four jobs in service class JUNK; but one is not be included in any count since it is not available for a WLM managed initiator. No job is not eligible for selection on all three MVS images. In service class JUNK, WLM sees three available jobs on SY1, zero available jobs on SY2, and one available job on SY3. Each WLM instance sees three jobs available for selection somewhere in the JESplex and 0 jobs unavailable for selection anywhere in the JESplex

  The count of jobs available for execution on a MVS image is adjusted up (more jobs enter the system destined to WLM managed initiators) or adjusted down (jobs are selected for execution or canceled).

  Similarly, service classes are registered as new work destined for a WLM initiator is encountered for a service class not currently registered. Service classes are also deleted if a extended period of time passes without any jobs waiting for initiation in that service class.

- WLM is not provided job backlog information by job class, job priority or any other traditional job attribute. These attributes play no role in work selection by WLM. They may have been considered when the job was classified by WLM services at the discretion of the creator of the service definition.

- Only JES knows which jobs are destined to WLM managed initiators and which systems these jobs are eligible for execution.

Job Selection

There are two distinct categories of initiators, those owned by JES and controlled by JES commands and those owned by WLM. A job can be selected by either a JES initiator or a WLM initiator but not both. JES can tell the difference between the initiator types and does not allow an initiator to acquire a job destined for the other category of initiator.

The following are the basics of JES2 job selection.

- JES initiators can select work from multiple job classes.
- Initiators from each JES instance within a JESplex select work from the common job queue independently of other initiators on other members of the JESplex.
- JES2 initiators select the oldest available work within priority within job class specification order.

Lets take an example of what this means. Suppose there are three jobs on the queue waiting for selection at time t4 (where t0 < t1 < t2 < t3 < t4).

- Jobclass: A

Jobname: MCCOX1	Priority: 7	Service Class: AA	Time on Queue: t1
Jobname: MCCOX2	Priority: 6	Service Class: BB	Time on Queue: t3

- Jobclass: B

    Jobname: MCCOX3   Priority: 14   Service Class: AA   Time on Queue: t2

Suppose an initiator named AB has job class list of (A,B). When the initiator comes looking for work, it attempts to find the oldest job in the highest priority for job class A, if no jobs are available it repeats the search with for job class B. If there are no available jobs it waits patiently until one appears and tries again. Thus the initiator selects jobs MCCOX1, MCCOX2 and MCCOX3 in that order.

Now suppose jobs in class A and B are to be destined to WLM managed initiators. WLM managed initiators only request work from one service class. JES supplies the oldest available job in the service class. If the initiator requests work from service class AA, it would select jobs MCCOX1 and MCCOX3 in that order. If the initiator requests work from service class BB, it would select job MCCOX2. If no job is available, the initiator waits until either a job becomes available or WLM withdraws the job select request.

It is possible that a JES managed initiator can request a job from a job class whose work is being sent to WLM managed initiators. JES does not satisfy the request. It appears to the initiator as if there is no work available in the job class. The initiator takes the normal action for a JES initiator as previously discussed.

JES and WLM managed initiators do not ask for a job by name or number. They ask for one from a category. JES must make the determination of whether a job that appears to be selectable is really available; if the job is not available, it is skipped and the next job examined. The important thing to remember is that many events can occur between the time an initiator makes a request to JES for a job and the request is serviced.

Information Returned on the Job Select.

JES provides a great amount of information to the initiator in response to a job select request, regardless of the type of initiator. In addition to the information returned previously, the following information is provided by JES on the job selection request and is reflected in the SMF30 records.

- Service classification
    - Current
- Job execution delays
    - All 4 categories of execution delay.

Queue Position

Jobs waiting for selection by WLM managed initiators are segregated by service class. Within the service class queue, jobs are ordered by age. More accurately, the jobs are ordered by the time they were made available for initiator selection. This time stamp is kept with the job even if it is moved from one service class queue to another as the result of an operator command. For example, consider the following:

There are two service classes: HOT and COLD. Jobs are placed on the execution selection queue at time ‘tx’, (t0 < t1 < t2 ...).

- HOT

MCCOX1	t1
MCCOX3	t3
MCCOX4	t4

- COLD

MCCOX2	t2
MCCOX5	t5
MCCOX6	t6

Now suppose the operator changes the service class of job MCCOX2 from service class COLD to HOT. JES inserts MCCOX2 into service class queue HOT using its original queued to execution time value ‘t2’. After the action is completed the queue of waiting jobs looks like:

- HOT

MCCOX1	t1
MCCOX2	t2
MCCOX3	t3
MCCOX4	t4

- COLD

MCCOX5	t5
MCCOX6	t6

Maintaining the original time stamp means a job is not sent to the end of the queue if its service class is changed. This also means the operator does not have the capability of moving a job to a more advantageous position in the queue of waiting jobs.

Operational Support

JES provides support for WLM managed initiators at the normal expected (or required) level. These enhancements allow the operator to continue using the JES interface to monitor and control the batch workload in a manner consistent with prior experiences. The standard expectations for JES function include:

- Configuration definition

  JES allows the specification of a job class (JES2) to be WLM managed in the initialization stream.

- Configuration altering

  JES allows the operator to change a job class (JES2) from JES management to WLM management (or vice versa) through appropriate JES specific operator commands.

- Job management

  JES allows the operator to set/change the service class associated with a job through appropriate JES specific operator commands.

  JES allows the operator to manipulate jobs in the backlog based on service class value. For example, all jobs of a given service class can be placed in operator hold.

- Job Inquiry

  JES allows the operator to display jobs in the backlog to determine their service class, or display a list of jobs in the backlog with a specified service class.

See the appropriate JES commands and messages manuals for details.

Demand Initiation

JES operators have always had methods of getting a job started in a JES initiator through some series of JES commands. There is no operator control over WLM initiators as there is with JES managed initiators. However, a means has been provided to force a job destined to a WLM managed initiator into execution. A new JES2 command $S J can be issued by the operator to force a job into execution. Command syntax does not allow a specific system to be designated for initiation. The target system can be left up to JES and WLM or the operator must set the system affinity mask to the system where the job is to be run prior to issuing the $S J command.

The $S J command does not override all ‘hold’ conditions on a job. The command honors duplicate job name hold, $PXEQ, and system affinity and resource affinity hold.   The command ignores job hold, job class hold and job class limits.

It is important to understand the ‘start job’ command merely allows the operator to force the job into a WLM managed initiator. There is no implied capability to get the job to run. Whether or not it receives machine resources (i.e., CPU and storage) is still under the control of WLM.

The ‘start job’ command is only applicable for jobs destined to WLM managed initiators. The command has no effect on jobs destined for JES managed initiators.

Limiting Initiation

JES operators have always controlled the number of jobs that are allowed to run in a given class on each MVS image of the JESplex. This was accomplished by limiting the number of initiators eligible to select jobs from the class. There is no corresponding external control mechanism over the number of WLM managed initiators actively selecting work from a specific service class or the total number of initiators running work in a service class.

When job classes are being used to control the level of simultaneous access to some resource (e.g. public work packs, storage, etc.) there remains the need to control the number of jobs executing simultaneously when the number of initiators is controlled by WLM. JES2 now allows the operator to specify the maximum number of jobs (i.e., XEQCOUNT) which can be active in a given class somewhere in the MAS. In JES3 terminology, JES2 has now implemented TDEPTH.

Job class limits may cause WLM to presented with an inaccurate view of the job backlog. For example, consider a single system JESplex where:

- Job class A jobs are destined for WLM managed initiators.
- The execution limit for job class A is 2. No class A jobs are currently in execution.
- There are currently 10 class A jobs in the queue, 5 jobs in service class HOT and 5 jobs in service class COLD.

When JES tells WLM of the job backlog how many jobs are available to run (or not run) in each service class? Is it

1. 2 jobs available in service class HOT; 3 jobs unavailable in service class HOT;
   0 jobs available in service class COLD; 5 jobs unavailable in service class COLD.
   - or -
2. 0 jobs available in service class HOT; 5 jobs unavailable in service class HOT;
   2 jobs available in service class COLD; 3 jobs unavailable in service class COLD.
   - or -
3. 1 job available in service class HOT; 4 jobs unavailable in service class HOT;
   1 jobs available in service class COLD; 4 jobs unavailable in service class COLD.
   - or -
4. 2 job available in service class HOT; 3 jobs unavailable in service class HOT;
   2 jobs available in service class COLD; 3 jobs unavailable in service class COLD.

The correct answer is #4. JES may over-indicate the availability of work for a given service class. When two jobs are in execution from job class A, regardless of service class, JES does not allow any additional jobs from the class to be selected by WLM initiators. Once 2 class A jobs are in execution, JES reports 0 jobs available in service classes HOT and COLD until one of them finishes.

System Dry Up

Occasionally MVS images must be taken down in the JESplex. In the past, JES commands were issued to dry up initiators, allowing them to finish processing the job they currently had but not allowing them to select the next job. There are no commands to tell WLM managed initiators to quiesce or dry up.

JES2 has a new command ($P XEQ) which causes all initiators to dry up on a specific member of the MAS. This command effects both JES and WLM managed initiators. Subsequently a $S XEQ command must be issued to allow job selection to occur on that MVS image.

WLM Responsibilities

The role of WLM remains the same, only its scope has increased with the new initiator management support. Essentially, WLM is now aware of job backlog and has the ability to affect the length of time a job is queued for execution. There are a large number of changes that must be incorporated into WLM, the SRM, the initiator, and the JES to fulfill the previous statement. These changes are categorized and described in the following sections.

JES Specific Services

The services basically fall into two areas, classification of batch work and JES-WLM collaboration.

Classification Services

WLM expects JES to classify arriving batch work before it is selected by the initiator. The initiator and other components have classified work for quite some time, so there is really nothing new in this area. If the JES level doesn't support work classification, the initiator continues to classify the batch job just as it has in the past. In fact, JES and the initiator pass the same information to the WLM classification services.

The service classification assigned to a batch job is based upon:

- Job name
- User id
- Job class
- Accounting information
- PERFORM= value (new with OS/390 R3)
- Job priority

Classification of work may occur multiple times prior to selection for initiator or during the execution phase at the discretion of the operator or installation processing specifications. JES does not invoke WLM classification services for TSO logons, started procedures or started jobs.

Collaboration Services

The majority of the changes to WLM and JES are in the collaboration area. WLM must be aware of JES environment changes (and vice versa) which include:

- A new JES instance appearing in the JESplex.
- A JES instance disappearing from the JESPLEX.
- A new service class associated with the batch environment appearing.
- A service class associated with the batch environment disappearing.
- A request from JES to initiate a job immediately.
- Activation of a new WLM policy.
- The number of jobs in the backlog within WLM's span of control.

These changes occur through one of two mechanisms.

- Explicit calls of WLM services invoked by JES to alert WLM of a change in the batch environment, a request of WLM or information on the state of the job backlog.
- ENF notifications, whereby WLM notifies JES of a change requiring action by JES.

These areas were discussed in the ‘JES Responsibilities’ section. Should you desire to write your own JES, consult the WLM management services guide and reference.

Initiator Control

While all this JES-WLM handshaking is all very interesting, the crux of the problem is how does WLM intelligently manage the starting, stopping and work selection criteria. More importantly, how does this affect the flow of jobs through the system.

WLM Initiators

WLM managed initiators and JES managed initiators differ in a few subtle ways. Once a JES managed initiator is started, it is essentially in one of two states: executing a job or requesting a job. The initiator doesn't really care what kind of job it receives from JES, JES code limits the selection scope for the initiator based on operator specified work selection criteria (i.e., the job class list, A and B for example). If no suitable work is available, the empty initiator always sits waiting for a job of the proper class to appear. The initiator waits until selectable work appears or the initiator is terminated. The operator can change the work selection criteria; but essentially the initiator either runs a job or waits patiently for work. The JES managed initiator selects work based on job class, it doesn't know about service classes and doesn't care whether work is being serviced promptly or appropriately. JES initiators are completely unaware of job backlog depth in making job selections.

WLM manages its initiators in a different manner allowing it to react to changes in workload mix, (both in execution and waiting for execution). Since WLM knows the job backlog mix, how well work is progressing through the sysplex and this system, the current system MPL, system constraints, etc., WLM can make an intelligent choice as to which category of work (i.e., service class) a job should be selected from or whether ANY job should be selected at all. WLM can and does change the service class selection value for its initiators at any time depending upon availability of work and system resources. An initiator may wait for WLM direction because there is no work to be selected or no additional work should be initiated. So, a WLM managed initiator can be in one of three states:

1. Running a batch job
2. Requesting a batch job from JES
3. Waiting for WLM to tell it what to do next

WLM initiators are started under the master scheduler (SUB=MSTR) similar to other system address spaces (e.g., LLA, VLF, IXGLOGR, etc.). WLM managed initiators do not require a JES job number. WLM uses the same INIT procedure that JES uses to start initiators. The initiator determines whether it was started by JES or WLM and acts appropriately.

Only WLM can start and stop WLM managed initiators. If the operator issues a ‘P INIT’ command against a WLM managed initiator, the initiator does indeed terminate. However, WLM detects this event and immediately starts another one in its place. WLM keeps track of the number of initiators it has started and the type of work they are processing, requesting to process, or sitting idle.

Conceptually, WLM maintains a pool of initiators. WLM initiator control for processing work is accomplished at two levels. Initiators can be started (added to the pool) or stopped (removed from the pool) as needed. The type of work selected by WLM managed initiators can be changed as conditions change for active (in the pool) WLM managed initiators.

WLM may or may not have complete control over the number of initiators processing jobs in a specific service class. JES managed initiators may be selecting work which is assigned to a service class which is being selected WLM managed initiators. This occurs because WLM initiators select by service class and JES initiators select by job class The installation can cause jobs in different job classes to be assigned to the same service class. WLM can only control the selection of jobs in its own initiators.

Adjusting the Number of Initiators for a Service Class

The number of address spaces (initiators) backing each service class is based on goal achievement, observed address space delays and the multi-programming level (MPL). Decreasing the number of address spaces is driven by the differential between the MPL and the number of address spaces; having address spaces consistently swapped out is the system is overloaded with more important work. Increasing the number of address spaces is driven by detection of the service class not meeting its goals and a significant portion of the observed delay is job initiation delay. Of course, the current MPL for the service class must be considered prior to increasing the number of address spaces.

When reducing the number of address spaces associated with a service class, WLM has two options. The first option is to find the address spaces waiting for JES to provide a job and recall them. The second option involves catching the initiator 'between jobs'; letting the initiator finish the job it is running, but preventing it from selecting another from the same service class. These initiator address spaces are returned to the pool where they can be used to process jobs associated with other service classes or terminated.

When increasing the number of address spaces associated with a service class, WLM again has two options. The first option is to assign unbound initiators in the pool to the service class. The second option is to create additional address spaces using the INIT procedure and associate them with the service class in need of additional address spaces.

Determining When/Where an Initiator Should be Started

WLM bases its decision whether to increase the number of initiators on several criteria. These include the availability of system resources (e.g., memory and CPU cycles), the availability of batch jobs and the importance level of the service class, and whether or not starting an initiator is projected to have a positive effect on reducing the execution delay portion of JES queue time. There are three cases for which WLM creates a new initiator to process work for a given service class.

Case i: A system in the JESplex is underutilized and there are jobs available for that system.

In this instance WLM has ample machine resources to apply to the jobs that are running on the MVS image. Jobs may not meet their goals; but it is not due to a shortage of CPU cycles or memory resources. WLM starts an initiator(s) to allow additional jobs in the backlog to be selected for execution. The number of initiators will be limited by the amount of machine resources and/or the number of jobs in the service class.

WLM does not prevent jobs from starting merely because the service class is over achieving its goals in this case. If there are more than one service class with work available, WLM attempts to keep all service classes over achieving its goals by two mechanisms that occur on an interval basis:

1. Evaluating which service class is least over achieving its goals and starting a single initiator for that service class.

2. Pacing the start of initiators for a service class over multiple evaluation intervals, allowing other service classes to add additional initiators.

Case ii: A service class is not meeting its goals due to 'waiting for initiation' delays

In this instance, WLM must determine whether or not starting some number of additional initiators is projected to have a positive impact on the service class meeting its goal and at what cost to other work running on the system. Additional initiators require CPU cycles and memory. WLM must determine the availability of these machine resources or find donors for these resources (i.e. work in execution in over-achieving service classes of the same importance or service classes of lesser importance). If there are no available machine resources, then WLM will not start additional initiators.

It is important to understand in this case, there must be a positive effect on the execution delay component of JES queue time as it relates to achieving the service class goal. For example, if service class OVERNITE is not meeting its goal because 5000 jobs are in the input queue with the same service class and there is sufficient resource to start only a single initiator, then WLM may choose not to start another initiator for that service class. This occurs when WLM algorithms predict that starting another initiator will not have significant impact on reducing queue delay and subsequently allow the service class to meet its goals.

Case iii: A service class has a queue of jobs available for only one system in the JESplex

In this instance, WLM realizes that some type of affinity is causing jobs to be bound to only one system and preventing these jobs from running on another system in the JESplex which is better suited from a memory and CPU cycles perspective. In this case, a minimum of one initiator is started on the system to service the queue of jobs for the service class. Thus jobs in the service class will progress through the system.

Requesting Work

Basically, a WLM managed initiator requests work from JES just like a JES managed initiator. The same mechanisms are involved, JES can determine whether the request came from a JES initiator or one of WLM's initiators. If there is work available for the initiator it is be returned to the initiator. If there is no work for the initiator, the initiator waits.

JES is obligated to keep batch jobs destined for WLM managed initiators separate from batch jobs destined to JES initiators. Batch jobs destined for WLM initiators are separated into service class queues and ordered within each queue by age on the queue. Queuing position is determined by the time of selection eligibility not the time the job entered the system. Job selection from a service class queue is FIFO (or age) under normal situations. Installation limits on the number of jobs of a given class which can be in execution affect job selection. For example, if the first job on the service class queue would cause installation job class limits to be exceeded, it will be not selected and the next job on the service class queue is potentially eligible for selection.

A JES initiator can be awakened with a job to process or a request to terminate. A WLM initiator can be awakened for a third reason; to return to WLM for new instructions.

Initiator Display

Neither JES nor WLM provide commands to display the state of WLM managed initiators. There is no mechanism to display the number of initiators currently running a given service class or waiting for work to appear in a service class.

Dirty Initiators

WLM managed initiators may suffer from the same storage fragmentation problems as JES managed initiators. WLM does not detect ‘dirty initiators’ and automatically replace them with a clean one. The P INIT,A=asid command can be used to terminate the dirty initiator. WLM detects this event and replaces the terminated initiator with a new one.

ENQ delays

JES2 managed initiators may select jobs which cannot be executed due to ENQ conflicts encountered by the initiator. WLM initiators are not immune to this condition either.

Canceling and requeuing a job that is delayed by ENQ conflicts has some interesting side effects. Please refer to the section in the Migration Considerations chapter.

Initiation Control

The initial reaction to WLM managed initiators seems to be

> *It's about time!*

followed by:

> *Oh no, I'll lose control over the batch workload!*

The intent of this section is to dispel the latter reaction.

The controls installations have had over batch jobs initiated by JES still exist for batch jobs initiated by WLM. In fact, there are new operational capabilities for WLM managed initiators which do not exist for JES managed initiators.

There are some subtle differences between JES and WLM management of awaiting batch jobs. We usually think of jobs in execution as being swapped in and running or swapped out and waiting to run, with the number in or out being determined by the MPL. Think of jobs waiting for WLM managed initiator selection as **jobs that haven't been swapped in yet, but waiting to run**.

Preventing Jobs from being Selected for Execution

Simply put, any reason preventing a job from being selected by a JES managed initiator prevents the job from being selected by a WLM managed initiator. Refer to section Job Delays Prior to Execution for the various reasons a job is not be selected by a JES managed initiator. All of these reasons apply to WLM managed initiators as well.

Holding and Releasing Job Classes

The previous paragraph is almost 'true'. In JES2 releases prior to OS/390 R4 (and for OS/390 R4 which has not yet been $ACTIVATEd), job classes can be held and released on a system basis. Once the $ACTIVATE command has been issued, job class attributes and settings are MAS-wide. This means that a job class is held on the MAS basis and not a system basis. This action is enforced for both JES2 and WLM managed initiators. The implications of this feature are: the installation cannot prevent jobs from being selected by WLM managed initiators on a designated system by holding the job class on that system. If the job class is held, no WLM initiator (or JES initiator) on any system is allowed to select work associated with the designated class.

WLM and JES2 together provide a mechanism to target jobs to a specific set of MVS instances through the use of Resource Affinity Scheduling. If it is necessary to prevent jobs in a job class destined to WLM initiators from executing on certain members of the MAS, the creation of appropriate scheduling resources and environments and their association with jobs in the job class could be implemented.

Allowing WLM Managed Initiators Access to Jobs in the Queue

The installation has total control over the categories of jobs which are destined to WLM managed initiators. The determination as to whether JES or WLM initiates a job is by job class. Either JES schedules or WLM schedules all the jobs in a class. At any time, the operator can change the job class between JES mode and WLM mode. This action affects all queued but not yet selected batch jobs as well as jobs not yet queued for initiator selection. JES will keep track of this in order that WLM and JES remain in synchronization. Therefore, it is quite easy to ‘test drive’ WLM managed initiators for various workloads and subsequently return to a previously understood environment if necessary.

Controlling the Number of Jobs in Execution

This facet provides the greatest operational difference. For JES managed initiators, the simplest way of limiting the number of jobs in execution for a specific class is to limit the number of initiators that can select jobs of that class. If you only want five jobs from job class A running on SY1, only start five initiators that can select jobs from job class A.

For WLM managed initiators the installation must use a different JES provided means for limiting the number executing jobs for a specific class. JES2 provides the ability to limit the number of jobs in execution on a job class basis throughout the entire MAS via XEQCOUNT. (Please refer to the section on Limiting Initiation in the prior chapter.) Using the XEQCOUNT feature is supported for job classes destined for both JES and WLM managed initiators.

Limiting the number of jobs in execution may require a change in operator procedures. The execution limits are applicable to both JES and WLM managed initiators.

Forcing a Job into Execution

The procedure for getting a job destined to a WLM managed initiator into execution is quite simple. (Please refer to the section on Demand Initiation in the prior chapter.) It is also different from the underhanded methods used by operators today (e.g., changing job classes, changing job priorities, starting initiators, etc.). The operator has to know which type of initiator the job is destined and perform the appropriate procedure. In either case, the end result (of forcing a job into an initiator) are the same: The job is initiated; but doesn’t' necessarily get the needed CPU resources to execute quickly.

Tracking Jobs Prior to Execution

SDSF View of Jobs Waiting for Execution

Fortunately, SDSF has been enhanced to assist operations and users in managing batch work in this new environment. From the ‘SDSF INPUT QUEUE DISPLAY’ or the ‘STATUS

DISPLAY' panels, a new action character displays detailed information for the designated job in a pop-up window.

The following figure illustrates the new pop-up window.

```
                          Job Information
Job name         MYJOBA    Job class limit exceeded? NO
Job ID           JOB01901  Duplicate job name?       NO
Job schedulable? YES       Time in queue             00:01:25
Job class mode   WLM       Average time in queue     00:12:05
Job class held?  NO        Position in queue         125   of 350
                           Active jobs in queue      5
Scheduling environment: PRIMESHIFT       available on these systems:
AQTS____  AQFT____  ________  ________  ________  ________  ________
________  ________  ________  ________  ________  ________  ________
________  ________  ________  ________  ________  ________  ________
________  ________  ________  ________  ________  ________  ________
________  ________  ________  ________  ________  ________  ________
________  ________  ________  ________  ________  ________  ________

F1=Help       F2=Split      F3=Cancel     F7=Backward   F8=Forward
F9=Swap      F12=Cancel
```

The panel displays the name of the required scheduling environment and members of the MAS on which the scheduling environment is currently available. None of the information in this pop-up window is modifiable. This panel is to display the reasons preventing a job from executing. There are other SDSF panels to delve into suspected scheduling environment configuration details.

Migration Considerations

This section describes the issues of getting from where we are to where we want to be taking into account WLM managed initiator features and eccentricities.

Multiple JESplexes

Multiple JESplexes of either or both flavors of JES can exist in a sysplex. There is no requirement that all JESplexes use WLM managed initiators. WLM is aware of the scope of each JESplex in the sysplex. However, it is important to remember the WLM policy is sysplex wide and independent of the JESplex or JES image.

From an operational control point of view, batch jobs are still managed at the JESplex level.

You should avoid the situation where there are two JES environments exist in the sysplex with different expectations for a specified service class. For example, a test environment and production environment in the same sysplex. If the same job is submitted in each environment, it receives the same service class assignment. WLM understands the sysplex wide definition and importance of the service class and doesn't care about JES boundaries and whether or not one is test and the other production.

Secondary JES and Poly-JES

WLM does not support secondary JESes in some configurations. Specifically, this means WLM does select batch jobs from a secondary JES in the same MAS as the primary JES. JES2 installations using ‘poly-JES’ must evaluate their use of batch in the secondary JES. If the job class is destined for JES managed initiators, the situation is business as normal. If the job class is destined for WLM managed initiators, jobs in the WLM designated class are not selected from the secondary JES.

JES2 has a concept of ‘master JES’ in the situation where the primary and secondary JES instances belong to the same MAS. If the primary JES is terminated, the secondary JES becomes the master JES allowing jobs destined for WLM initiators to continue to be selected. Whenever, the primary JES returns, it assumes the role of master JES and WLM no longer selects jobs from the secondary JES.

If there are two JES2 instances on the same MVS image belonging to different MASes, WLM distinguishes between them and selects jobs from each of them.

Switching Between Goal Mode and Compatibility Mode

The simple answer is don't do this if you have any job classes being serviced by WLM initiators. The WLM initiators cease to select work from the job queue and terminate.

Similarly, while in compatibility mode, the operator should not change the mode of a job class from JES management to WLM management. No work can be selected as there are no WLM managed initiators allowed in compatibility mode.

Job Aging

Each JES has provided mechanisms to raise the priority of jobs as they age waiting for initiator selection. The intent was to keep the job from languishing forever in the input queue as newer, higher priority work entered the system. Some installations use this feature while others don't. Jobs awaiting selection by WLM managed initiators do not have their priorities adjusted as they age, even if the installation continues to age jobs waiting for JES managed initiators.

JES2 Eccentricities

When priority aging occurs in JES2, the job's priority is actually changed. This aging process causes JES to invoke the WLM classification services and may result in the job being assigned to a new service class. At any rate, a great number of jobs have the potential to be re-classified. Jobs awaiting selection in WLM managed initiators are exempt from priority aging process. The installation should carefully evaluate whether this aging process is to their benefit.

Changing a Job's Priority or Service Class

Traditionally, operators have used a JES command to raise the job's priority when it was waiting for execution, in an effort to have JES schedule the job sooner by moving it closer to the head of the job queue. For jobs destined to WLM managed initiators, this action may not have the desired effect. Jobs scheduled for a WLM initiator are not ordered by priority, only by the time they were first assigned to a service class queue. Changing the job’s priority causes JES to re-classify the job. This priority change can lead to either the service class being left as it was before, or a new service class assigned. In either case, the position on the service class queue is not affected by changing its priority.

Work in service classes processed by JES and WLM initiators

The installation has the option of continuing with JES managed initiators in addition to WLM managed initiators. It is highly likely that jobs destined to both types of initiators may be assigned the same service class value. Once jobs are in their execution phase, they are treated the same whether they reside in WLM or JES managed initiators. If there are a great number of jobs being processed through the system in JES initiators and the goals for the service class are being met or not being met, one may observe unexpected delay for jobs destined to WLM managed initiators relative to JES managed initiators. This is because WLM has no influence over JES placing jobs into execution in JES initiators.

Batch Job Submission Methods

Batch queue management is most important for production batch schedules that must be done in a certain time window or time critical batch which affects some online system's real-time response (e.g., quick data extract and report). There are two extreme positions in batch submission. These methods have an impact on WLM’s selection of work based on execution queue delay.

Just in Time Submitted Batch

One of the attributes of scheduler submitted jobs is they are seldom submitted before their predecessors complete. In fact, in systems that whose initiators are well managed, they seldom accumulate any execution delay time. To prevent elongation of elapsed times, jobs in the critical path require aggressive service class definitions and assignment. Jobs not on the critical path should be examined to determine whether they should be in the same service class.

Mass Job Submissions

On the other extreme are the installations that dump hundreds or thousands of jobs into the system at the beginning of the batch cycle and let JES and MVS sort everything out and run the jobs in priority order as resources permit.

Managing Queue Times

JES queue time plays a role in both response time goals and velocity goals. Goal attainment has a direct influence on the number of WLM managed initiators actively supporting a service class. So it is important to understand how JES queue time (and the execution delay component in particular) influences WLM measurements of goal attainment.

Know of the jobs waiting for initiation may lead to WLM increasing the number of initiators, resources permitting, in an attempt to meet installation defined goals. However, as previously pointed out WLM may predict that adding initiators for the service class will not significantly reduce the initiation delay portion of JES queue delay.

Response Time Goals

For jobs in execution in a service class having a response time goal, the actual computed JES queue time, as described previously, is used in the assessment of goal attainment. The elimination of user delay from JES queue time may warrant the re-evaluation of service classes with response time goals. Job classes that were not suitable for response time goals due to high percentage of TYPRUN=HOLD jobs may now be better candidates.

Velocity Goals

The execution delay portion of JES queue time is now a component of the velocity calculation. Velocity calculations are based on sampling the state of all jobs in a service class that are in their execution phase or are waiting for initiator selection. (To reiterate, one should consider jobs waiting for an initiator to select them as jobs that ‘have not been swapped in yet’ and not noticeably different from jobs in execution that are swapped out.) The number of jobs eligible for selection are counted in the denominator of the velocity calculation the same as jobs that are swapped out or queued for CPU. Jobs in resource hold, operational hold, etc. are not counted in this calculation.

This change in velocity calculation can have dramatic affects on the flow of jobs in the installation depending on how batch is submitted. If you refer to the previous two types of production job submissions. There is little impact on the velocity calculation for the ‘just in time submission approach’ as there is not a great number of jobs in the queue just waiting for an initiator. (Similarly, a great numbers of jobs that are not schedulable due to resource unavailability, job class limits, duplicate job names, or other operational delays does not impact the velocity calculation.)

On the other hand, is the mass submission of jobs which are only prevented from running by the availability of an initiator. In this case, a large number of jobs in the queue can significantly bias the velocity calculation and thus goal attainment.

Note: It is important to understand that queue time predictions for a service class is based on jobs that WLM knows are in the queue. If a job is destined to a JES managed initiator, it is not counted in the velocity calculation.

Setting Goals for Batch Jobs

NOTE: This section was added 10/15/02 on the following page.

Removing WLM Managed Job Classes from JES2 Initiator Lists

This is not a requirement but is recommended just to reduce confusion among operators and casual users of SDSF.

Setting Goals For Batch Jobs

As indicated previously, the starting of new WLM initiators is predicated on the batch service class missing it's goal due to JES queue delay ***and*** WLM making the determination that starting new initiators will reduce this delay and thereby contribute to the batch service class meeting the installations defined goals.
Before establishing or re-establishing goals for WLM batch initiator candidates the first item to undertake is to understand your environment, that is, analyze your batch job mix:
Look at the SMF 30 records - job distribution by class, by shift, ET, ...
Look at the SMF 72 records - multiple periods, DPRTY, queue delays, velocity, response times, and the performance index if reported.
Look at your current job class rules and how they are assigned and enforced.

Second, analyze your initiator structure and ask yourself these questions:
Do you have any asymmetric configurations ?
Which job classes are restricted to specific systems ?
Are there any special limits by jobclass or by system ?
Are there any limits on work packs, tape drives, storage, etc. ?
Do you make changes via operator procedures or automation ?

With a good understanding of your batch environment you can begin to evaluate your goals and determine which service classes might be better served by WLM initiators. Keep in mind that when using WLM managed initiators, JES queue time is now a part of the velocity calculation as it is with the response time calculation. Also TYPRUN=HOLD is no longer a factor in response times. The list below attempts to establish some guidelines as to whether response time objectives or velocity goals would be better suited for a service class. The actual velocities and response times used for your batch service classes will of course depend on their business objectives and the relationship with other workloads executing in the sysplex.

Good Candidates for Response Time Goals
Medium to high frequency jobs (at least 3 jobs ending every 20 minutes)
Jobs with steady arrival rates (not bulk submission by schedulers)
Short-running jobs
Jobs with uniform run times
Jobs where the resource delay impact on response times is minimal

Good Candidates for Velocity Goals
Long running jobs (queue time is not significant)
Highly variable execution times
Jobs with few completions (not enough historical data to establish a response time)

Non-Candidates (Keep these with JES-managed Initiators)
Jobs to run immediately or emergency jobs
Just-in-time jobs managed by a job scheduler
Critical path job streams
Jobs with long resource delays
Jobs with system affinity
Operator held jobs
Tape-constrained environments, or jobs subject to waiting on dataset enqueue's
Jobs used to support IMS/CICS regions

Frequently Asked Questions

Q: Since service classes are sysplex wide entities, how should I have separate service classes for batch work that lives in different JES complexes cohabitant in the same sysplex?
A: By definition, if you use the same "job" in either JESplex you will get the same service classification since service classification rules are the same across all members of the sysplex. If you want them to be treated differently, they must be classified into distinct service classes based upon some criterion.

Q: In a multi-CPU environment, how are the job delays tracked since a job may be delayed on one machine for reason A and on a different machine for reason B?
A: JES has a complex wide view of delays for each job. If a job is eligible to run anywhere in the sysplex, the job is considered eligible to run. If a job is not eligible and the only reason is that it requires an abstract resource that is not available in the sysplex, the job is counted as delayed for resource affinities. Otherwise the job is counted as operationally held. JES passes queue information to WLM that allows WLM to have a sysplex wide (which may include multiple JESplexes) view of batch work and delays.

Q: If a service class is over-achieving its goal on a processor and there are batch jobs awaiting execution in this service class will WLM start additional initiators?
A: It depends. If there are available resources to start the initiators and the resources are not needed by other work not achieving its goals then WLM will start another initiator, The logic is similar to the logic WLM uses to raise MPL when the machine is under-utilized.

Q: Can an operator issue a command to stop or quiesce WLM from selecting additional work in a given service class on a specific MVS instance?
A: No. Work selection is controlled at the individual job level, the job class level or the entire system level.

Q: How can an operator limit the number of jobs executing in a service class?
A: The operator cannot control the number of jobs executing in a service class directly. Through JES commands an operator can stop any jobs from executing on a specific system which would stop WLM from putting any initiators on that system. However it is all or nothing: If the operator allows WLM to put initiators on a system, only WLM controls the number of initiators on that system.

Q: How can an operator cause WLM to limit or spread its initiators to specific MVS instances?
A: The operator cannot control the number of jobs executing in a service class directly on any member of the JESplex. Through JES commands an operator can stop any jobs from executing on a specific system which would stop WLM from putting any initiators on that system. However it is all or nothing: If the operator allows WLM to put initiators on a system, only WLM controls the number of initiators on that system.

Q: How can an operator determine why a job has not been selected for execution by WLM?
A: The JES flavor of spool browser has been enhanced to display the reason a job is not selectable (e.g. operational hold, job class limits, resource affinity hold, etc.) as well as its position on the queue.

Q: What happens if a MVS image is taken out of goal mode on one member of the JES complex?
A: Nothing for JES managed initiators. WLM managed initiators will cease to select work and terminate.

Q: When a system joins the sysplex and JES complex how quickly will WLM populate the image with initiators?
A: It depends. If batch is missing its goals because of lack of initiators and the new system has lots of spare capacity, WLM will start adding initiators as soon as it sees the system. If batch is meeting its goals and the only reason to add new initiators is that the new system is under-utilized and there are queued jobs, WLM will add initiators slowly.

Q: When a lot of work is dumped into JES, how quickly will WLM respond and start additional initiators?
A: It depends on available capacity. If there is available capacity, the under-utilized code will kick in within 10 seconds. If there is not available capacity but WLM projects that batch will miss goals without more initiators, WLM will take resource from other work to start initiators if it makes sense.

Q: How does WLM detect/handle dirty initiators?
A: It doesn't detect dirty initiators. The installation will have to detect job failures due to storage abends, find the offending initiator and cancel it. WLM will replace the canceled initiator.

Q: How long does WLM leave an initiator empty before changing its selection criteria or terminating it? Will this cause thrashing or unnecessary overhead?
A: If there is only one initiator in the sysplex for a service class, WLM will leave it associated with the service class for at least an hour after the queue becomes empty. Additionally WLM will terminate initiators if the number of initiators is more than 1.5 * the long term average number jobs in the class. WLM will also terminate initiators if there is not the CPU or storage available to support the jobs that are running. In this case if there are no free initiators will be terminated as jobs end.

Q: Does this mean that we should consider reducing the number of job classes?
A: This depends on the use of the job class. If the job class is being used to assign attributes, probably not. If the job class is being used for scheduling control, then perhaps resource affinity scheduling feature is a better, more flexible mechanism.

Q: Should we re-consider how many job classes get mapped into a service class or how a single job class gets mapped into multiple service classes?
A: The WLM management of initiators should not cause an installation to have to re-consider how many job classes get mapped into a service class. The one exception to this is that mixing JES managed job classes and WLM managed job classes in the same service class might cause problems. See question below. If a job class has a limit on the number of jobs, splitting it across service classes may cause WLM to start unnecessary initiators. See question about limits below.

Q: If the job has its priority or job class changed by the operator will it be re-classified using the WLM services? Will the original and current classification be available to installations in exits and/or SMF data?
A: If a job has its priority changed it will be reclassified using the WLM classification service. The old service class will not be available to exits but will be available in SMF26 data and an indicator is set in the SMF30 records indicating the job has been reclassified.

Q: Is there an JES or WLM operator command to set the service class of the job while it is waiting for execution, similar to the execution time service?
A: Yes. There are JES commands to explicitly set the service class of jobs awaiting initiator selection. This command can be used on jobs destined for either JES or WLM managed initiators.

Q: When there is a limit on the number of jobs of a given class that can be active in the JESplex (JES2) are the counts of available jobs for a service class adjusted down, or does JES just skip over the jobs of that class when WLM initiator does a job select? The job is not really in a hold condition as it is a JES scheduling delay.
A: WLM sampling sees the jobs above the job class limit as being ineligible for execution. WLM keeps an average of the number of jobs eligible to run in the service class. Jobs above the job class limit are not part of that average which stops WLM from starting initiators that cannot be used because of the limits. Using an MPL analogy these jobs are not part of the "ready user average". One note is that the WLM sampling code only understands a job class limit within a service class. If a job class with a limit is split across service classes, WLM might start extra initiators thinking more jobs are eligible to run than really are.

Q: Is there an exit point where the installation can influence which job in the service class is chosen by WLM during job selection? Is this a JES or WLM exit if it exists?
A: JES2 will provide an exit for WLM managed initiator selection similar to EXIT 10 for JES managed initiator selection. The exit will be allowed to veto a job selection only.

Q: Can the production control packages assign the service class to the job at submission time, rather than have JES invoke the classification services?
A: No.

Q: Does WLM keep initiators for the service classes it manages waiting for a job, or does a job have to show-up before an initiator is started.
A: WLM will not start an initiator until the first job for a service class is available for execution.

Q: What prevents 1 job from being submitted and WLM on each member of the JES complex from starting an initiator to select the job?
A: The WLMs on each system communicate with each other. Once a system starts an initiator, it broadcasts that fact to all the other systems in the sysplex. It should be unlikely that 2 systems would start initiators at exactly the same time. On each system WLM will only be considering starting initiators once every 10 seconds.

Q: Is there a JES or WLM command to show the number of initiators requesting work from a service class? Are there any commands to display the number of jobs active in WLM managed initiators?
A: There is no command to display the number of empty WLM managed initiators and the service class for which they are selecting work. There are JES commands that allow you to display jobs by service class.

Q: The definition of execution delay queue time will change with this level of OS/390 to reflect the time the job could be selected not the total time in the system. Will this have any affect on or require changes to my current policies?
A: Batch response time is being changed to not include time jobs spend in a TYPRUN=HOLD or TYPRUN=JCLHOLD state. This is not expected to impact current policies because we believe the use of response time goals for batch service classes is limited to jobs that have little queue time.

Q: How does Batch Accelerator feature of Smart Batch work with WLM managed initiators.
A: Basically, Batch Accelerator works the same for jobs that run in WLM managed initiators and JES managed initiators. Job steps that are selected to run in parallel are placed into Smart Batch initiators. Smart Batch makes the determination as to which system in the JESplex the individual steps are executed.

Q: So what does job priority mean when a job is waiting for initiator selection? Does priority affect selection within a service class?
A: For jobs destined to WLM managed initiators, priority is not considered when queued for execution. Priority was used when the job was classified into a service class originally or when an operator modified the priority.