WLM:
Resource Affinity Scheduling

OS/390 R4

Document Date: August 22, 1997

Trademarks

The following are trademarks of the IBM Corporation:

- OS/390
- CICSplex/SM
- CICS
- IMS

Abstract

This document describes the resource affinity scheduling function introduced by the Workload Manager function of OS/390 R4 and exploited by the JES component in OS/390-JES2 R4 for batch job scheduling.

Reader Responsibilities: This document is a work in progress. The reader is obligated to make constructive comments on content, style, organization, areas needing additional explanation, etc.. Please send your comments to Mike Cox at any one of the following addresses:

- MCCOX at WSCVM
- USIBMWGT at IBMMAIL
- coxm@us.ibm.com

If you wish to send a fax, the number is: 8/372-8818 or 301-240-8818.

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
David Bostjancic	WLM Development, S/390 Division
Greg Dritschler	WLM Development, S/390 Division
Bill Keller	SDSF 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

Special Notices

The information contained in this document has not been submitted to any formal IBM test and is distributed on an ‘as is’ Basis **without any warranty either expressed or implied**. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead. Any speculation by the author about possible implementations of WLM functions in IBM products is purely conjecture and does not imply any commitment by IBM.

Using Resource Affinity Scheduling

This chapter is intended to provide some insight into the requirement for resource affinity scheduling and the approach taken by the Workload Manager component and other components in providing a solution.

Overview

A single OS/390 system provides a great number of diverse **application environments** with its myriad of subsystems, data base managers, devices, languages, etc.. Combining a number of systems into a sysplex allows an installation to provide an even greater number of application environments. One of the main usage issues encountered by the user and systems management issues facing the installation is:

> *not every system is capable of providing all application environments at all times.*

An application environment may be limited to a subset of systems within the sysplex due to software licensing, machine resources, peripheral device attachments, data base accessibility, etc.. Additionally, an application environment may be available only during certain times of the day or week or when other application environments are not enabled.

Installations use a number of means to prevent the user from having to know exactly where and when an application environment is available. In the case of batch jobs, job classes, job priority, independent software vendor (ISV) products, and locally developed code are used to control where and when a job is executed to satisfy its execution environment requirements. The submittor of the job may be required to explicitly state his application environment requirements, the installation may discover the requirements, or some combination of both. By hiding the intricacies of the underlying structure, the installation has greater flexibility in how it provides a service. Changes to the computing environment can be made without adversely effecting the users.

The Resource Affinity Scheduling (RAS) function provided by the MVS Workload Manager (WLM) extends the capability of workload scheduling components (e.g., JES) by enhancing their scheduling mechanisms. Use of resource affinity scheduling is *optional* and under complete control of the installation. Since resource affinity scheduling is a function of WLM, it is a requirement that there be an active WLM policy. However, there is *no requirement for WLM goal mode.*

The installation can define multiple environments to support applications in terms of installation provided resources and schedule the execution of work according to the availability of the required environment. Specifically, the new WLM function allows an installation to:

- define **resource elements** within the sysplex,
- set the **state** of each resource element independently on each system,
- define **scheduling environments** comprised of one or more resource elements, with required states, representing a set of execution time requirements, and
- provide information on the state of the resource elements and the availability of the scheduling environments to interested parties (e.g., JES, Automation) which make decisions on whether or not the required execution environment is available on an MVS instance within the sysplex.

A **resource element** is a representation of an execution time resource. The installation must identify and name these entities. A resource element can be a physical entity, such as a data base, a peripheral device, a machine feature, etc.. However, a resource element can also be more abstract, such as ‘second shift’, ‘over the weekend’, ‘cheap cycles’, etc.. Normally, each resource element represents one execution environment characteristic which must be considered when scheduling work.

A resource element always has an assigned **state** of “ON”, “OFF” or “RESET” to indicate the intended use or presence of the resource. A resource element such as ‘SHIFT’ may have “ON” (representing ‘Prime Shift’) and “OFF” (representing ‘Non-Prime Shift’) states. In this instance the use of ‘SHIFT’ resource is to delineate less busy periods of the day from extremely busy periods of the day. The changing of ‘SHIFT’ from ‘Prime Shift’ to ‘Non-Prime Shift’ state (and vice versa) doesn't necessarily have to occur at the same time each day or simultaneously on all systems in the sysplex.

A **scheduling environment** is a grouping of some number resource elements, each element having a specified required state, into a single named entity to be used in making scheduling decisions. The scheduling environment is either available (i.e., all the resource elements are in the required state) or unavailable (i.e., one or more of the resource elements are not in the required state) on an MVS image. When a scheduling environment is associated with a unit of work, it signifies the requirement for an execution environment providing ALL of the services/conditions needed for successful execution.

Conceptually, resource elements are the building blocks used to define the various environments (i.e., scheduling environments) throughout the sysplex.

A resource element exists in one and only one state on each MVS instance within the sysplex; however, they may exist in different states on different MVS instances within the sysplex. Thus, scheduling environments are made available or unavailable on a particular MVS instance by manipulating the state of the resource elements on the MVS instance.

Resource elements may be specified in multiple scheduling environments with the same or different states. For example, one could create two scheduling environments: ‘prime time scientific’ and ‘cheap time scientific’ using resource elements: ‘vector facility’ and ‘shift’. The ‘vector facility’ would be designated in the same state for both scheduling environments. However, the ‘shift’ resource element would be specified in different states in these two scheduling environments.

One can have multiple scheduling environments available on each MVS instance. The availability of a scheduling environment on a MVS instance is solely dependent upon the state of the resource elements on the MVS instance. For example, the ‘DB2_PROD’ and the

‘DB2_TEST’ scheduling environments could be available on same, different, all, or none of the MVS images within a sysplex.

Throughout this document the terms **unit of work** and **execution scheduler** are used by design instead of “job” and “JES”. The WLM services for resource affinity scheduling are available to all work schedulers. You can easily substitute ‘batch job’ for ‘unit of work’ and ‘JES’ for ‘execution scheduler’ if you prefer to think in those terms. However, do not think of this facility as applicable in only the ‘JES’ framework.

Scheduler responsibility

It is the responsibility of the execution scheduler (e.g., JES3, APPC, etc.) to place a unit of work into execution on any MVS instance of the sysplex within its scope. Each scheduler incorporating resource affinity scheduling capability into its existing mechanisms must:

- Provide a means to associate a valid scheduling environment with the unit of work.
- Interrogate WLM to determine if the named scheduling environment is available on an MVS image.
- Schedule the unit of work into execution appropriately.

WLM does not provide a mechanism to attach a scheduling environment to a unit of work. This process is specific to the execution scheduler (e.g., JES, CICS, etc.). WLM only provides management and interrogation services for resource elements and scheduling environments. Hence resource affinity scheduling is a collaborative effort by the execution scheduler and WLM.

The first exploiter of resource affinity scheduling is the JES component for batch job scheduling. Resource affinity scheduling augments the current JES mechanisms for limiting the execution phase of a job to the correct or desired system. The use of scheduling environments reduces the need for additional job classes or specification of system affinity. MVS customers currently have batch workloads requiring services available only on a subset of images within the sysplex. For example, jobs may require access to:

- Vector Facilities for execution.
- A data base manager which does not support sharing.
- A data base manager in a data sharing group which does not include all MVS instances in the sysplex.
- A data base which must be off-line to transaction managers in order to perform some utility function.
- A non-shareable device such as an optical storage device, Open Systems Adapter, etc..
- A subsystem existing on only a limited number of MVS instances.
- A compiler existing on only a limited number of MVS instances.

A JES Example

The resource affinity scheduling capabilities and limitations are illustrated with the following example.

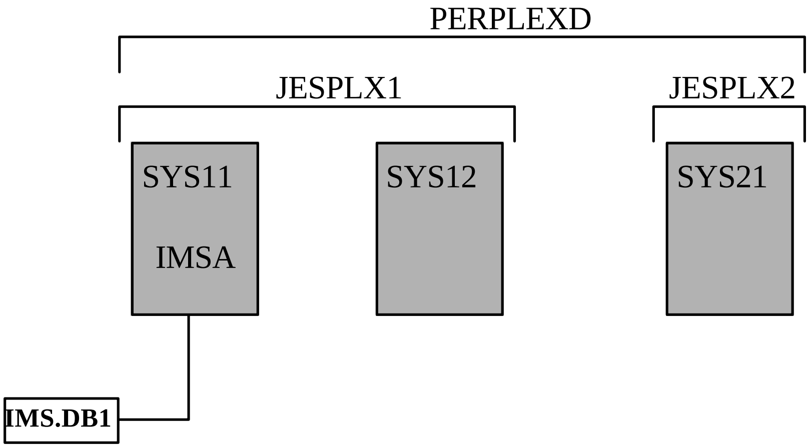
Consider the sysplex ‘PERPLEXD’ containing MVS instances

- ’SYS11’,
- ’SYS12’ and
- ’SYS21’.

Within the sysplex, there are two JES complexes:

- ’JESPLX1’, containing systems:
  - SYS11, and
  - SYS12
- ’JESPLX2’, containing system:
  - SYS21

The installation runs an IMS subsystem, ‘IMSA’. There exists in IMSA an application which utilizes database ‘IMS.DB1’. This IMS subsystem runs in JESPLX1, normally on system SYS11. The following figure illustrates the environment.



In this simple, hypothetical example, the data base, IMS.DB1, may be accessed by some BMPs while online access is occurring, whereas other BMPs must only access the database while online processing is suspended.

A mechanism is required which allows JES to distinguish jobs in the two categories of BMPs in order to allow execution at the proper time on the correct MVS instance. WLM and JES resource affinity scheduling enhancements provides the installation with a mechanism to control where these BMPs run and when they are allowed to start. The following steps illustrate how this would be accomplished.

1. The installation defines the resource scheduling environment through extensions of the WLM ISPF dialogues. First the resources themselves are defined and then the scheduling environments are defined.

   A. **Define two resource elements** to WLM:

      - ’RSC_IMSA’, represents the IMSA subsystem.

      - ’RSC_IMS.DB1’, represents the database

      RSC_IMSA in the ON state indicates the presence of the IMSA subsystem. The OFF state is not explicitly given meaning in this example, but implies IMSA is not present on the system.

      RSC_IMS.DB1 in the ON state indicates activity by online systems; whereas, OFF indicates there is no online activity against the data base.

   B. **Define two scheduling environments** to WLM specifying the previously defined resource elements and the required states:

      - ’ENV_DB1_ONLINE’, indicating the data base is available to online transactions. This environment requires:

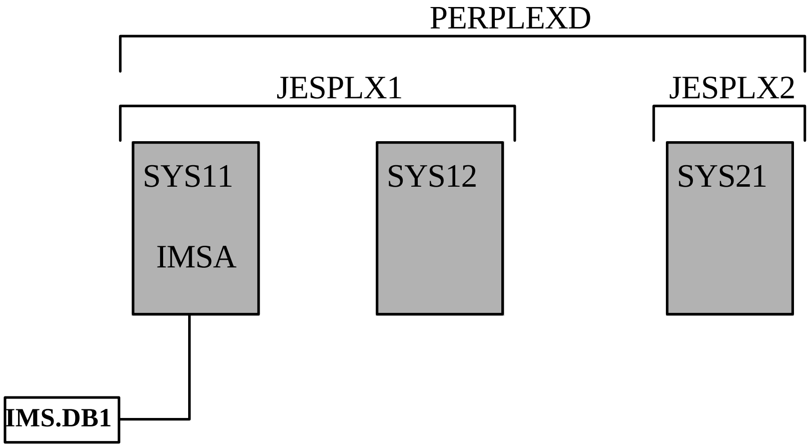
        RSC_IMSA in the ‘ON’ state, and
        RSC_IMS.DB1 in the ‘ON’ state.

      - ’ENV_DB1_OFFLINE’, indicating the data base is not available to online transactions. This environment requires:

        RSC_IMSA in the ‘ON’ state, and
        RSC_IMS.DB1 in the ‘OFF’ state.

2. The **WLM policy is activated** on any system in the sysplex. Knowledge of the resource elements and scheduling environments are propagated to each WLM instance in the sysplex, and all newly defined resource elements are placed in a ‘RESET’ state. The RESET state is an unschedulable state which cannot be specified in a scheduling environment. All scheduling environments requiring a resource element in the ON or OFF state are unavailable until action is taken to set the state of the underlying resources to a schedulable state (i.e., ON or OFF). All jobs requesting a scheduling environment not available on any system will be placed in a ‘scheduling environment hold’ condition. This prevents jobs which require a scheduling environment from untimely execution during transition periods.

The following figure illustrates the situation at this stage.



Resource Elements:			
RSC_IMSA	RESET	RESET	RESET
RSC_IMS.DB1	RESET	RESET	RESET
Scheduling Environments:			
ENV_DB1_ONLINE	Unavailable	Unavailable	Unavailable
ENV_DB1_OFFLINE	Unavailable	Unavailable	Unavailable

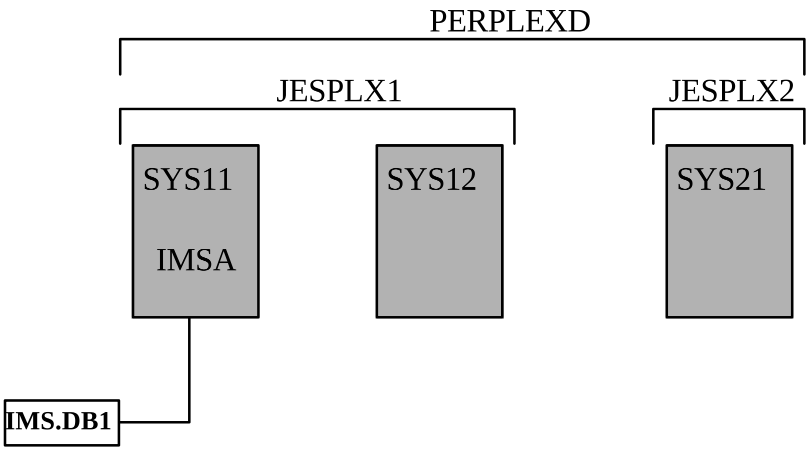
3. With **IMSA active only on SYS11 and allowing access to IMS.DB1 from online transactions,** the state of the two resource elements should be **modified to WLM** so that:

   - RSC_IMSA is set to ‘ON’ on system SYS11, (indicating the IMS subsystem is active on this system), and left in the ‘RESET’ on systems SYS12 and SYS21.
   - RSC_IMS.DB1 is set to ‘ON’ on SYS11 (indicating the database is in the interactive state as opposed to the batch state), and left in the ‘RESET’ on systems SYS12 and SYS21.

   Based on our definitions of the scheduling environments and the setting of the two resource elements on SYS11, this action would result in scheduling environment:

   - ENV_DB1_ONLINE being available only on SYS11.
   - ENV_DB1_OFFLINE being unavailable on all systems in the sysplex.

The following figure illustrates the situation at this stage.



Resource Elements:			
RSC_IMSA	ON	RESET	RESET
RSC_IMS.DB1	ON	RESET	RESET
Scheduling Environments:			
ENV_DB1_ONLINE	Available	Unavailable	Unavailable
ENV_DB1_OFFLINE	Unavailable	Unavailable	Unavailable

At this point, scheduling environments care established and may be used in the proper scheduling of batch jobs.

4. A **BMP job, 'BMPJOBA', is submitted in JESPLX1.** This BMP must execute using IMSA and concurrent with online activity against IMS.DB1. BMPJOBA is given a scheduling environment requirement: ENV_DB1_ONLINE.

   BMPJOBA is eligible for initiator selection on system SYS11, but has a 'scheduling environment hold' condition on system SYS12, as its scheduling environment is available only on SYS11. JES and WLM will determine BMPJOBA can be selected for execution only on the SYS11 image.

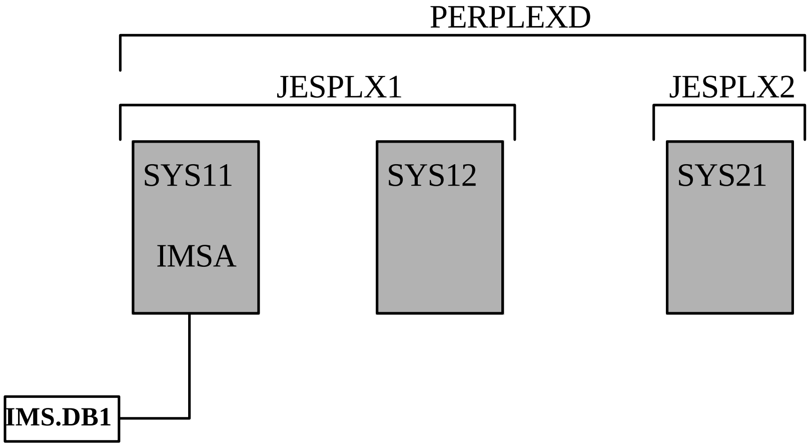
5. A second **BMP job, 'BMPJOBB', is submitted in JESPLX1.** This BMP must execute using IMSA while there is no concurrent online activity against IMS.DB1. BMPJOBB is given a scheduling environment requirement: ENV_DB1_OFFLINE.

   BMPJOBB is in a 'scheduling environment hold' condition on both SYS11 and SYS12 and is not eligible for initiator selection on either system, as its scheduling environment is unavailable on both systems in JESPLX1. There may well be an initiator available on SYS11; however, JES and WLM will together determine that BMPJOBB cannot be selected for initiation.

6. When **online access to IMS.DB1 is stopped, a WLM modify command** would set the RSC_IMS.DB1 resource element to the 'OFF' state on SYS11.

At this point, the resource requirements for scheduling environment ENV_DB1_OFFLINE are met on SYS11, and the resource requirements for scheduling environment ENV_DB1_ONLINE are no longer met on SYS11. (Nothing has changed on SYS12 or SYS21.)

The following figure illustrates the situation at this stage.



Resource Elements:			
RSC_IMSA	ON	RESET	RESET
RSC_IMS.DB1	OFF	RESET	RESET
Scheduling Environments:			
ENV_DB1_ONLINE	Unavailable	Unavailable	Unavailable
ENV_DB1_OFFLINE	Available	Unavailable	Unavailable

7. **JES will detect** the availability of scheduling environment ENV_DB1_OFFLINE and the unavailability of scheduling environment ENV_DB1_ONLINE. The job, BMPJOBB, is now eligible for initiator selection on system SYS11. BMPJOBB still has a ‘scheduling environment hold’ condition on system SYS12, as its scheduling environment is available only on SYS11. Again, JES and WLM will together determine that BMPJOBB can be selected for execution only on the SYS11 image.

Several aspects of resource affinity scheduling, some of which are not obvious, should be noted from the above example.

- The state of all resource elements and the availability of all scheduling environments are known throughout the sysplex. Each WLM knows the state of resources and scheduling environments on every MVS image in the sysplex. WLM keeps this information synchronized across the sysplex.
- The MVS instance chosen for execution by JES was not determined by a special job class or a designated system affinity. The appropriate time for execution on the correct MVS instance was not determined by removing a JCL hold, a job class hold, or a priority hold condition. None of the traditional JES mechanism preventing a job from entering execution were used to control when and where these BMPs executed. JES is scheduling jobs based on the availability of the job’s resource requirements.

- Jobs identify their resource requirements by specifying a single scheduling environment. Jobs do not refer to resource elements and required states directly.
- Operators control the resource element states on each system; they do not directly control the scheduling environments. Any resource element can be set to one of three states: RESET, ON, or OFF by the operator. WLM then reflects the resource element state changes in the availability or unavailability of scheduling environments throughout the sysplex based on the definitions in the WLM policy.
- A single resource element may be referenced in multiple scheduling environments. Setting the state of a single resource element may make multiple scheduling environments available or unavailable on a MVS instance
- The scheduler (JES) discovers and reacts to scheduling environments becoming available or unavailable on the various systems. The scheduler doesn't need to be aware of the underlying resource elements and their states.
- Jobs waiting for execution are not bound to a system. If IMSA is moved to system SYS12 in JESPLX1 and the resource element states are correctly re-specified then the affected scheduling environments will be correctly established. Jobs requiring those scheduling environments will then be eligible for selection on the appropriate system.
- The resource elements, or the required state of an resource element, comprising a scheduling environment can change without awareness by the submittor of the job or the JES at anytime prior to selection of the job by an initiator.
- WLM provides no interface to alter the scheduling environment associated with a job. In fact, WLM does not know which jobs have a scheduling environment requirement. Only the scheduler knows this information.
- While resource elements and scheduling environments are known throughout the sysplex, there may be other conditions which constrain where a job executes. In our example, if IMSA is moved to SYS21, BMPJOBA and BMPJOBB cannot be scheduled there because SYS21 is not a member of the JES complex in which the job was submitted.
- WLM and JES do not control access to these resource elements and scheduling environments via System Authorization Facility (SAF). Resource affinity scheduling is a scheduling mechanism to get the job to the correct execution environment. Normal security mechanisms should be used to control access to the real resources represented by a scheduling environment.
- The IMS instance known as IMSA does not have any knowledge of the installation defined resource elements known as RSC_IMSA or RSC_IMS.DB1. Definition and manipulation of the resource elements on any given MVS instance must be done outside of IMS.
- A scheduling environment which is unknown to WLM may be associated with a job. It is up to the JES to determine what to do with the job (e.g., hold it on some error queue, delete the job, etc.) depending on when this condition is discovered. A scheduling environment could be deleted from the active WLM policy while a job is waiting for the scheduling environment to become available.

The previous example is a JES example, as JES is the first component to exploit the Resource Affinity Scheduling function provided by WLM. There is no implicit or explicit restriction of this function to the JES component. For example, CICSPlex Systems Manager can be extended to understand and use these constructs when determining appropriate CICS

AORs for transactions. In the prior discussion, one could replace ‘job’ with ‘unit of work’ and ‘JES’ with ‘work scheduler’ and all statements remain true.

What’s in a name

The name, Resource Affinity Scheduling, has special significance. It is meant to restrict the scope and expectations for this facility. In essence, this feature allows installations to direct units of work to the proper execution environment. Specifically, resource affinity scheduling does **not** do:

- Complete abstract resource scheduling

  This is not an attempt to solve nor implement a unified abstract resource scheduling mechanism. There is no concept of shared/exclusive use of a resource. Similarly, there is no concept of resource quantity or consumable/reusable resources.

  Additionally, the implementation does not and can not replace JES3 device setup (a specific manifestation of the general problem).

- Dynamic definition/deletion of resources

  Resource elements and scheduling environments are ‘statically defined’ through the WLM administrative utility. The static list can be altered by activation of a new WLM policy at any time. However, there is no mechanism for a unit of work (or a scheduling system) to dynamically create a resource, manage its state, have other related units of work ‘use the resource’ and subsequently delete the resource following successful completion of all related units of work.

Rationale for Scheduling Environments

As the previous example illustrates, the execution time requirements for a unit of work are indicated via an indirect manner. First, the installation defines resource elements to WLM. Then, scheduling environments are defined specifying a set of resource elements with required states which will exist in the computing environment. These scheduling environments, along with their installation specific meanings and usage rules, are then published for use by users of the computing facility. A unit of work requiring a specific execution environment would then have one of the installation defined and managed scheduling environments associated with it.

While batch exploitation is eminent and the constraints of job control language cause usability issues, JES is not the only scheduler needing resource affinity scheduling capability. It is important to remember resource affinity scheduling capability is available to units of work other than batch jobs. There were two options considered for the specifying the resource requirements associated with a unit of work.

> **Option 1**: Allow the user (or subsystem) to associate a list of resources and desired states with the unit of work.

Option 2: Allow the installation to predefine named lists of resources/states which are valid within the installation and allow the specification of one named item with a unit of work.

Option 1 presented a number of obstacles:

1. Each scheduler must manage inconsistent requirements. If ‘JOE’ is a resource and the user specifies LIST=(JOE(ON),JOE(OFF)) requesting resource ‘JOE’ in the ON state and resource ‘JOE’ in the OFF state.

2. There is no means to prevent a user from requesting a resource requirement set which will never exist in the operating environment. If ‘SYS1’ and ‘SYS2’ are resources representing different MVS systems respectively; then LIST=(SYS1(ON),SYS2(ON)) could never be satisfied as a MVS instance has only one identity.

3. Each scheduler must implement mechanisms for managing some number of resources. The ‘correct number’ being defined by each scheduler and/or installation.

4. Each scheduler must implement a means whereby a user or administrator could specify the resource requirements to the scheduler (i.e., a control language extension).

5. Any extensions to resource definition and/or management made by WLM would require replication within the various schedulers. Any extensions to resource requirement specification would require changes (consistent?) to each scheduler.

   Current implementation of resource affinity scheduling requires all resource elements to be in the requested state for the scheduling environment to be considered available. Extending the capability to support other logical operators (e.g., ((IMSA == ON .OR. IMSB == ON) .AND. IMS.DB1 == ON)) would require changes to each scheduler.

6. Defining the usage rules in a clear concise manner to casual users of the scheduler.

7. Handling the introduction, and more importantly, the deletion or change in context of resource elements without disruption of work flow (e.g., not having to change old existing JCL to accommodate a change).

Option 2 offered solutions to a number of issues raised in **Option 1**:

1. Valid groupings of resource elements and states into scheduling environments are under the control of knowledgeable personnel in the installation. The casual user cannot create ‘never will occur requirement lists’; he can only choose from a valid predefined set of scheduling environments.

   It can be argued this implementation prevents a user from spontaneously creating a new list of resource requirements without consulting the administrator of the WLM policies, resulting in additional bureaucracy. The counter argument goes like: it is highly unlikely a user will successfully generate a list of required resources and states which can be satisfied. Without a great deal of knowledge of the installation's policy for definition and management of resources and their states, the user will likely create a resource list which prevents the unit of work from scheduling as desired, if ever.

2. The scheduler does not require change to extend either the WLM management of resource types (e.g., consumable and reusable resources) or implement extensions to requesting resources (e.g., ((ONE == RESET .OR. TWO == RESET) .AND. .NOT. (THREE == RESET))). The scheduler has merely one named entity, which it requests WLM to determine whether it is available or not available on a specific MVS instance. The execution scheduler has no idea of the constructs or logic behind the service assessing the availability of the scheduling environment.

3. Resource elements may be added/deleted/modified from the scheduling environment without requiring a change to an existing unit of work resource requirement specifications. No JCL change is required.

4. Specification of a single entity greatly reduces the code and processing required by the scheduler.

5. Specification of a single entity greatly reduces the complexity of describing its usage to the casual user of the scheduler.

The chosen implementation of resource affinity scheduling incorporates resource elements and scheduling environments exposing only the scheduling environments to the schedulers and their users. This implementation provides tremendous flexibility to IBM and the installations using resource affinity scheduling. The installation can choose to have a scheduling environment associated with a single resource element (and state) or a large number of resource elements.

Rationale for Resource Elements with Multiple States

On each MVS instance, a resource element can exist in one of three states: RESET, ON, or OFF. A resource element may exist in only one of these states. Two states (i.e., ‘ON’ or ‘OFF’) are considered to be valid states for scheduling purposes, thus a resource element can only be requested to be in one of two states, either ‘ON’ or ‘OFF’, in a scheduling environment.

There is one state (i.e., ‘RESET’) which is considered to be invalid for scheduling purposes, and thus cannot be requested in a scheduling environments. WLM intentionally provides no capability to set the initial state of a resource element to either of the scheduling states through policy activation. The newly defined resources will be set in the RESET state. All scheduling environments referencing the resource element will be unavailable until the resource element is set to a valid scheduling state. This action will prevent untimely scheduling of units of work into execution prior to the time when the services they require are truly available.

There is no meaning implied by the name of the states ON and OFF. The states mean whatever the installation wants them to mean. Just as the resource elements are abstract in nature, so are these two states. The states could just as well have been called ‘A’ and ‘B’.

What is important to remember is which states are schedulable (i.e., ON and OFF) and which state is not schedulable (i.e., RESET). One must recognize the distinction between the ‘requested state of the resource element’ and the ‘current state of the resource element’ and when the requested resource element requirement is considered satisfied.

Table 1 Resource Requirement Satisfied

		Current State		
		ON	**OFF**	**RESET**
Requested state	**ON**	YES	NO	NO
	OFF	NO	YES	NO

For a scheduling environment to be considered available, each of its resource element requirements must be met according to the above table.

Allowing resource elements to exist in one of these three states allows the installation to:

1. Conserve or reduce the number of resource elements needed to provide adequate scheduling capability. Additional resource elements are required in many cases when only two states (i.e. ‘ON’ or ‘OFF’) are allowed. One state is invariably required to support a ‘cannot or do not schedule condition’ specification.

   For example, consider the resource element named ‘SHIFT’. Using ON to denote ‘Prime Time’ and OFF to denote ‘Non-Prime Time’, one can use a single resource element and a single operator command to allow jobs to run during prime time or non-prime time. If only an ‘ON’/’OFF’ state were allowed, one would have to define two resource elements: ‘Prime_Shift’ and ‘Off_Shift’ and appropriately enable and disable these resource elements at the proper times.

2. Less operational complexity as there are fewer commands to issue in the correct order, and fewer opportunities to have ‘out of synch’ situations occur.

3. Ability to easily prevent jobs from executing on a specific MVS instance in the complex by setting/leaving the resource in a RESET state.

There are numerous uses for resource elements having one of two mutually exclusive scheduling states. There are also resource elements which conceptually have only one scheduling state such as a hardware feature which is either present or not. There may be requirements for resource elements with more than two mutually exclusive scheduling states or more than two non-mutually exclusive states. The installation has the flexibility to define and control additional resource elements and use them in a manner meeting these requirements if needed.

RESET state after IPL

When a system is IPLed, all defined resource elements are placed in the RESET state for the system. WLM does not remember the state of the resource elements from the point at which the system was last active and restore them. There is no WLM policy definition mechanism to set the resource element to a predefined state.

This action is intentional. While the prior state could have been remembered, WLM cannot determine at what point in time, after the IPL, the resources should be set to their previously existing state. Setting resource element states to a pre-existing or pre-defined state at an inappropriate time may cause units of work to fail as the resource is not truly available. The decision was made to make all scheduling environments unavailable by setting all resource elements to the RESET state until such time as the installation took explicit action to restore the scheduling environments. This action is best done through the automation instance which is restoring system service based on type of system start (e.g., normal, post-failure, etc.), time of day, etc..

WLM Services

WLM has a number of responsibilities in the implementation of resource affinity scheduling. WLM provides services to:

- define resource elements and scheduling environments,
- set the state of a resource element,
- inquire on the status of resource elements and scheduling environments, and
- provide sysplex wide coordination of resource element states and scheduling environments with resource schedulers.

Most importantly, all of these tasks must be done in a manner maintaining a consistent view across all members of the sysplex.

Definition of Resource Affinity Scheduling Entities

Definition of resource elements and scheduling environments is accomplished via new WLM administration panels.

Resource Elements

Resource elements must adhere to the following rules.

1. Limited to 16 character names. Each resource element may have an optional 32 character description visible through the WLM administration panel, specific inquiry commands and programming interfaces.

2. Limited to a total of 999 within the sysplex.

3. Must be unique within the sysplex.

4. There are no pre-defined resource elements. However, resource element names beginning with the characters ‘SYS_’ cannot be defined. (The “SYS_” prefix is reserved for IBM.)

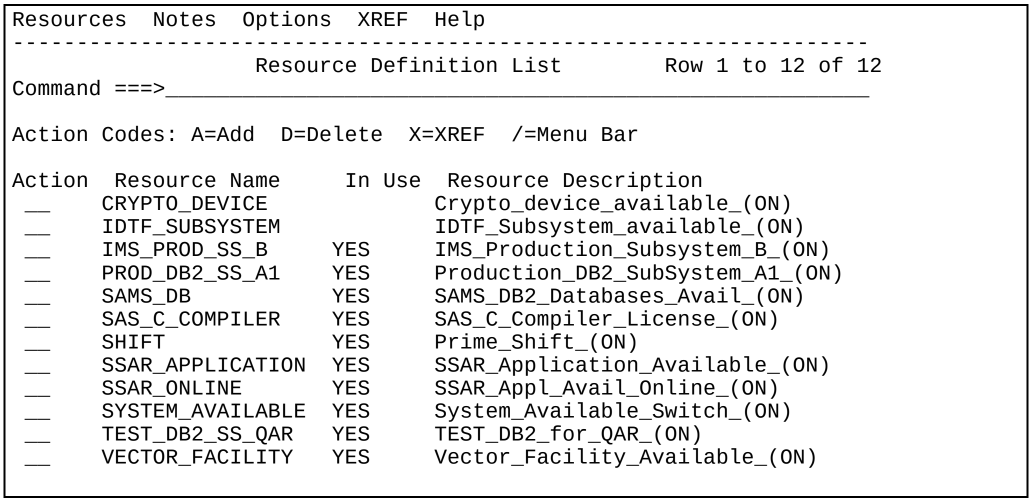
5. Are implicitly defined in the RESET or ‘unschedulable’ state.

   There is no capability to set the initial state of a resource following policy activation on a MVS image. The resources will be set to the RESET state, which will prevent untimely scheduling of units of work prior to when the service is truly available.

6. Do not have to be referenced by an scheduling environment.

7. May be referenced by multiple scheduling environments with the same or differing states.

The following figure illustrates the WLM ISPF panel used to define resource elements



```
Resources  Notes  Options  XREF  Help
-------------------------------------------------------------------
                   Resource Definition List        Row 1 to 12 of 12
Command ===>_______________________________________________________

Action Codes: A=Add  D=Delete  X=XREF  /=Menu Bar

Action  Resource Name     In Use  Resource Description
 __    CRYPTO_DEVICE              Crypto_device_available_(ON)
 __    IDTF_SUBSYSTEM             IDTF_Subsystem_available_(ON)
 __    IMS_PROD_SS_B      YES     IMS_Production_Subsystem_B_(ON)
 __    PROD_DB2_SS_A1     YES     Production_DB2_SubSystem_A1_(ON)
 __    SAMS_DB            YES     SAMS_DB2_Databases_Avail_(ON)
 __    SAS_C_COMPILER     YES     SAS_C_Compiler_License_(ON)
 __    SHIFT              YES     Prime_Shift_(ON)
 __    SSAR_APPLICATION  YES     SSAR_Application_Available_(ON)
 __    SSAR_ONLINE        YES     SSAR_Appl_Avail_Online_(ON)
 __    SYSTEM_AVAILABLE  YES     System_Available_Switch_(ON)
 __    TEST_DB2_SS_QAR    YES     TEST_DB2_for_QAR_(ON)
 __    VECTOR_FACILITY    YES     Vector_Facility_Available_(ON)
```

Several features of the panel require explanation.

1. Resource elements are defined with a name and an optional description.
2. Resource elements are kept sorted in ascending order.
3. Resource elements currently referenced by a scheduling environment are noted.
4. The XREF action displays all scheduling environments which reference the resource element (not shown).
5. Deletion of an resource element currently referenced by a scheduling environment is detected and not allowed.
6. There is no specification of a ‘state’ on this panel, the required state is set on the scheduling environment definition. Do not confuse the ‘(ON)’ or ‘(OFF)’ in the description field. This is just a hint to operations as to what a setting should be.

Scheduling Environments

Scheduling environments must comply to the following rules.

1. Limited to 16 character names. Each scheduling environment may have an optional 32 character description visible through the WLM administration panel, specific inquiry commands and programming interfaces.

2. Limited to a total of 999 within the sysplex.

3. Must be unique within the sysplex.

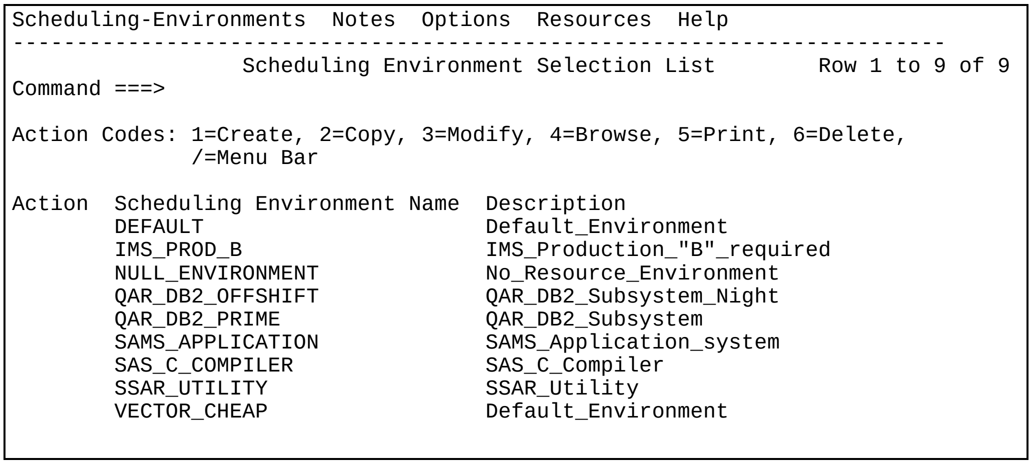
4. There are no pre-defined scheduling environments. However, scheduling environment names beginning with the characters ‘SYS_’ can not be defined. (The “SYS_” prefix is reserved for IBM.)

5. Can contain 0 to 999 resource elements, each with a specified state of either ON or OFF.

   Note: A scheduling environment with zero (i.e., 0) resource elements is available on all systems at all times. A scheduling environment no resource elements allows an installation to delete all resources from a scheduling environment, thus not have to set/reset resources.   No JCL change is required to jobs to remove references the scheduling environment. The jobs will behave as if no scheduling environment is specified.

There are two ISPF panels used to examine and manipulate the scheduling environments.

The following figure illustrates the panel used to define the scheduling environment.

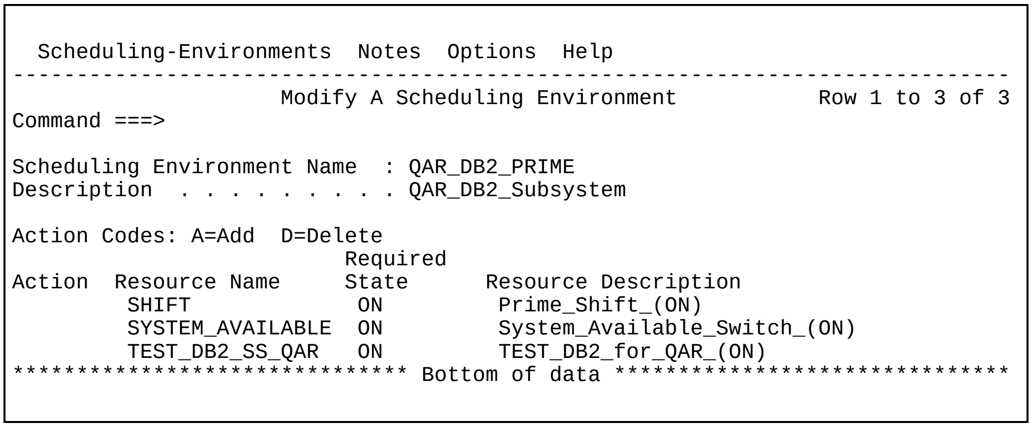


```
Scheduling-Environments  Notes  Options  Resources  Help
------------------------------------------------------------------------
                  Scheduling Environment Selection List        Row 1 to 9 of 9
Command ===>

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
              /=Menu Bar

Action  Scheduling Environment Name  Description
        DEFAULT                      Default_Environment
        IMS_PROD_B                   IMS_Production_"B"_required
        NULL_ENVIRONMENT             No_Resource_Environment
        QAR_DB2_OFFSHIFT             QAR_DB2_Subsystem_Night
        QAR_DB2_PRIME                QAR_DB2_Subsystem
        SAMS_APPLICATION             SAMS_Application_system
        SAS_C_COMPILER               SAS_C_Compiler
        SSAR_UTILITY                 SSAR_Utility
        VECTOR_CHEAP                 Default_Environment
```

Several features of the panel require explanation.

1. Scheduling environments are defined with a name and an optional description.
2. Scheduling environments are kept sorted in ascending order.
3. There is no specification of state for any resource element on this panel.
4. Deletion of a scheduling environment is allowed, whether or not it contains any resource elements or whether or not it is currently being referenced.
5. Creation of a scheduling environment referencing no resource elements is allowed.

Specifying the required resource elements with their desired states for a scheduling environment, is accomplished via a subordinate panel, as illustrated in the following figure.



```
  Scheduling-Environments  Notes  Options  Help
------------------------------------------------------------------------------
                     Modify A Scheduling Environment           Row 1 to 3 of 3
Command ===>

Scheduling Environment Name  : QAR_DB2_PRIME
Description  . . . . . . . . . QAR_DB2_Subsystem

Action Codes: A=Add  D=Delete
                          Required
Action  Resource Name     State      Resource Description
         SHIFT             ON         Prime_Shift_(ON)
         SYSTEM_AVAILABLE  ON         System_Available_Switch_(ON)
         TEST_DB2_SS_QAR   ON         TEST_DB2_for_QAR_(ON)
******************************* Bottom of data *******************************
```

Several features of the panel bear explanation.

1. Resource elements are listed in ascending order with optional description.
2. Each resource element specified must have been previously defined.
3. Each resource element specified must have a assigned state of ON or OFF.
4. A resource element can appear only once in the list.
5. Resource elements can be added/deleted or have their state changed independent of any other scheduling environment.

Resource Element and Scheduling Environment Coordination

WLM has the responsibility of maintaining a consistent view of the resource elements and scheduling environments across the sysplex. WLM must cope with the activation of new policies, the comings and goings of MVS images within the sysplex, etc., alerting schedulers of changes in the environment.

Activation of a New WLM Policy

Activation of a new WLM policy in the sysplex can cause a number of conditions to occur which must be handled properly by each WLM instance in the sysplex. The following conditions must be detected:

- new, previously undefined resource elements,
- previously defined resource elements,
- deleted (by omission) previously defined resource elements,
- new, previously undefined scheduling environments,
- previously defined, unmodified scheduling environments,
- previously defined, modified scheduling environments, and
- deleted (by omission) previously defined scheduling environments.

The key point is:

> *WLM, on each MVS instance, has complete knowledge of every resource element, with state information, and scheduling environment availability for every MVS instance in the sysplex. Furthermore, the information is consistent across the members of the sysplex.*

Conceptually, each WLM instance maintains a set of tables, one for each member of the sysplex. These tables contain the global definitions of all resource elements and scheduling environments, the state of all resource elements on each system, and the availability of all scheduling environments on each system.

Activation of a WLM policy requires WLM on each MVS image to reconcile the resource elements and scheduling environments defined in the newly active policy with resource elements and scheduling environments defined in the previously active policy. WLM does the following in reaction to a policy change affecting scheduling environments:

1. Identify and register each previously undefined resource element and set its state to RESET to prevent unexpected scheduling of units of work.

2. Identify and register each previously existing resource element whose usage is to be retained, leaving its state as it was prior to activation of the new policy.

3. Identify and de-register deleted resource elements. WLM does not know whether any unit of work has an implicit reference to the resource element through a scheduling environment.

4. Identify and register each previously undefined scheduling environment and set its availability or unavailability based on the list of resource elements and their states as established above.

5. Identify each previously defined scheduling environment and set its availability or unavailability based on the list of resource elements and their states as established above.

6. Identify and de-register each previously defined scheduling environment which has been deleted by omission.

7. Once a consistent sysplex-wide view of defined resource elements and scheduling environments has been established, an ENF(41) signal is issued on each system in the sysplex. This signal is an alert to interested parties of a major change in the resource affinity scheduling environment.

   Note: There is no ENF(57) notification for each scheduling environment that has become available or unavailable as the result of a policy activation. ENF(57) is only issued when a scheduling environment becomes available or unavailable due to a resource element changing state.

8. Create a SMF90 sub-record containing information for the currently active resource elements and scheduling environments.

System Removal

A system failure or sysplex partitioning action requires all remaining instances of WLM to set all the resource elements to the RESET state for the now absent system. Subsequently, no scheduling environment is available on the absent system.

System Introduction

System initialization, for a new or restarted MVS instance in the sysplex, is handled without benefit of memory of the past. WLM goes through the policy activation process on the initializing MVS system. All resources are discovered as new, not previously existing and placed in the RESET state on the MVS instance being initialized.

WLM on the initializing MVS system acquires the state of the resource elements and scheduling environments availability/unavailability status of other members of the sysplex in order that its view of resources in the sysplex is complete. Similarly, all other WLM instances in the sysplex detects a new WLM instance and mark all resource elements as being in the RESET for that MVS instance.

Resource Element State Modification

Setting the resource element to the appropriate state on each system is done by operator commands entered manually, through a programmed operator using the MGCR(E) service or using the IWMSESET service.

A new system-level command:

```
MODIFY WLM,RESOURCE=name,ON|OFF|RESET
```

is provided by WLM to set the state of the resource element. Execution of the command causes WLM to:

1. Identify and update the state of the designated resource element appropriately.
2. Propagate the new state information to other WLM instances.
3. Each WLM instance reflects this change in the appropriate scheduling environments.
4. A new ENF(57) signal is issued to alert interested parties in a change of state for each newly available or unavailable scheduling environment on the system that received the command. The ENF(57) is not issued on all the other MVS instances in the sysplex.

The operator does not directly control the availability of a scheduling environment on a MVS instance. The availability of a scheduling environment is predicated on all of its specified resource elements being in the correct state. Operators exert control over the state of the individual resource elements.

For example:

```
F WLM,RESOURCE=CASH_AOR,ON
IWM038I  12.21.05  WLM DISPLAY 181
  RESOURCE = CASH_AOR
  DESCRIPTION= ATM CICS Region
  SYSNAME    STATE
  SY1        ON
```

Scheduling Environment and Resource Element Inquiry

Each unit of work utilizing resource affinity scheduling must have a scheduling environment associated with it. The name of the scheduling environment is what the submittor of the unit of work and the scheduler use to identify resource affinity scheduling criteria. WLM understands the underlying resource element states for each scheduling environment. WLM provides commands to determine the root cause of a unit of work not being scheduled. WLM provides display commands which allow the operator to:

- List one or all scheduling environments defined in the sysplex, with descriptions, and the systems where the scheduling environment is enabled.

  ```
  DISPLAY WLM,SCHENV=*|scheduling_environment
  ```

  For example:

```
D WLM,SCHENV=CRYPTO
IWM036I  12.21.05  WLM DISPLAY 181
  SCHEDULING ENVIRONMENT: CRYPTO
  DESCRIPTION: Crypto required
  SYSTEM LIST:  SY1  SY2  SY3  SY5
```

```
D WLM,SCHENV=*
IWM036I  12.21.05  WLM DISPLAY 181
  SCHEDULING ENVIRONMENT: CRYPTO
  DESCRIPTION: Crypto required
  SYSTEM LIST:  SY1  SY2  SY3  SY5
  SCHEDULING ENVIRONMENT: DB2_TEST
  DESCRIPTION: DB2 test required
  SYSTEM LIST:
  SCHEDULING ENVIRONMENT: OFF_SHIFT
  DESCRIPTION: Non-Prime Time access
  SYSTEM LIST:  SY2  SY3
```

- List the details of a single scheduling environment on a specific system, indicating all the required resource elements with their states. If a scheduling environment is unavailable,

```
DISPLAY WLM,SCHENV=scheduling_environment,SYSTEM=system_name
```

For example:

```
D WLM,SCHENV=DB2PRODCCONFIG,SYSTEM=SY1
IWM037I  12.21.05  WLM DISPLAY 181
  SCHEDULING ENVIRONMENT: DB2PRODCONFIG
  DESCRIPTION: Product Required
  SYSNAME:      SY1
                   REQUIRED    CURRENT
  RESOURCE         STATE       STATE
   PRIMESHIFT      ON          ON
   DB2_PROD        ON          ON
```

```
D WLM,SCHENV=CRYPTO,SYSTEM=SY1
IWM037I  12.21.05  WLM DISPLAY 181
  SCHEDULING ENVIRONMENT: CRYPTO
  DESCRIPTION: Crypto required
  SYSNAME:      SY1
                   REQUIRED    CURRENT
  RESOURCE         STATE       STATE
   CRYPTO_HDWARE   ON          ON
  *DB2_PROD        OFF         ON
```

- List the state of one or all resource elements on one or all instances in the sysplex.

```
DISPLAY WLM,RESOURCE=*|resource_element[,SYSTEMS|SYSTEM=system_name]
```

For example:

```
D WLM,RESOURCE=*,SYSTEMS
IWM038I  12.21.05  WLM DISPLAY 181
  RESOURCE = DB2_PROD
  DESCRIPTION= DB2 Production
  SYSNAME    STATE       SYSNAME    STATE    SYSNAME   STATE
  SYS21      ON          SYSB       RESET    SYSC      OFF
 RESOURCE = CASH_AOR
  DESCRIPTION= ATM CICS Region
  SYSNAME    STATE
  SY1        ON


D WLM,RESOURCE=CASH_AOR,SYSTEM=SY1
IWM038I  12.21.05  WLM DISPLAY 181
  RESOURCE = CASH_AOR
  DESCRIPTION= ATM CICS Region
  SYSNAME    STATE  SY1        ON
```

WLM APIs for Resource Affinity Scheduling

WLM provides the following programming services which allows a scheduler (or other interested party) to determine the state of resource elements, set the state of resource elements, determine the validity of scheduling environments, and determine the availability of scheduling environments. All IWMSExxx services require the user to be in supervisor state or have a program key mask (PKM) in the range of 0-7.

WLM maintains sysplex-wide resource state information and scheduling environment availability information on each MVS instance. Therefore, one can acquire information from one WLM instance to make scheduling decisions on any other MVS instance in the sysplex. API's include:

- IWMSEDES **-** WLM Scheduling Environment: Determine Execution Service

  Provides the caller with information on the availability of a specified scheduling environment *for* any specified system *from* any system in the sysplex.

- IWMSEQRY **-** WLM Scheduling Environment: Query Service

  Provides the caller with information on scheduling environment(s) and/or resource element

  - Scheduling environment

    Provides detailed information about a specified scheduling environment, including all resource elements with their required states.

  - Resource element

    Provides the state of a resource element on a designated system from any system in the sysplex.

- IWMSEVAL **-** WLM Scheduling Environment: Validate Service

  Provides the caller with information on the validity of a specified scheduling environment from any system in the sysplex.

- IWMSESET **-** WLM Scheduling Environment: Set Resource Element Service

  Allows the caller to set the state of an resource element to the desired state. This service must be invoked on the system on which the resource element state is to be changed.

- ENF(41)

  Allows the listener to determine that a WLM policy activation has occurred which could impact work having a resource affinity scheduling dependency.

- ENF(57)

  Allows the listener to determine that a scheduling environment is now available or unavailable as a result of a resource element state change. This ENF is not a multisystem ENF, it must be listened for on each system in the sysplex within the execution scheduler's span of control.

SMF

SMF99 records contain new sub-sections that allow the installation to track the usage of scheduling environment at the installation level (i.e., SMF99) and at the job level as a result of changes to the SMF26and SMF30 records.

Beginning with OS/390-JES2 R4, JES2 tracks job queue delays prior to execution. There are four categories of delays which JES maintains. JES passes the accumulated times to the initiator for inclusion into the SMF30 records. One of the categories of delay is resource scheduling delay.

For jobs running in a JES2 environment, this is the amount of delay the job incurred when no system having the required scheduling environment available.

Batch and Resource Affinity Scheduling

This section details the mechanism whereby the JES component exploit the resource affinity scheduling capability provided by WLM. Resource affinity scheduling is only provided for batch jobs. There is no support for either TSO logons, started jobs or started tasks. If a scheduling environment is assigned to any of these types of work, it will be ignored.

WLM/JES/Converter Roles

Implementation of resource affinity scheduling for batch jobs is a collaborative effort among three components:

- WLM has responsibility of providing the interface to define the resources, manage states and provide inquiry functions, etc. as outlined previously.
- The MVS converter allows specification of a new keyword on an existing JCL statement to identify the scheduling environment request and verify its existence.
- JES continues to manage the jobs prior to execution and determine on which MVS instance jobs have their scheduling environment requirements met. JES allows jobs to enter the execution phase only when the scheduling environment is available.

The following sections provide detail on the responsibilities of each component and the interfaces required to provide resource affinity scheduling in the JES batch environment.

Scheduling Environment Specification

JCL

A new **optional** keyword (SCHENV=) is allowed on the job card to indicate the required scheduling environment needed for proper execution. For example:

```
//MCCOXA   JOB (C003,6363),'Mike Cox X-8588',
//             MSGLEVEL=(1,1),
//             REGION=4096K,
//             CLASS=A,
//             SCHENV=MAGIC1073,   <= scheduling environment
//             MSGCLASS=O   ...
```

Usage rules:

1. The scheduling environment name is limited to 16 characters. Any name containing characters other than 0-9, A-Z, #, @, $ must be enclosed in quotes.
2. One and only one scheduling environment may be specified for a job.
3. The scheduling environment name must be valid or a JCL error will occur.
4. Specifying a null scheduling environment (i.e. SCHENV=,) is not allowed and will result in a JCL error.

Converter and Generic JES

Converter Processing

The desired scheduling environment is declared by a keyword on a JCL statement. The converter must parse the keyword, determine if the associated keyword value is syntactically correct and validate the supplied scheduling environment using the IWMSEVAL service. If the supplied scheduling environment is not known to WLM, the job will be given a JCL error and terminated.

JES extracts the scheduling environment by examining the CI text created for the job. The scheduling environment is not propagated to any SWA control block and is thus undetectable by examining the SWA at job or step initiation. The installation has the option of modifying the scheduling environment in the JES internal text exit. The installation must assume responsibility for providing a valid scheduling environment. Converter validation of the specified scheduling environment as it appears in the internal text occurs after the JES internal text exits have been called. Refer to the JES2 Exits section for details on supplying and/or changing the scheduling environment for a job.

Note: Jobs specifying a //DD SUBSYS=xxxx must be converted on a systems where an instance of the named subsystem exists. The subsystem must participate in the conversion process. The converter/JES implementation of scheduling environments for batch jobs does not allow the direction of jobs to a converter on a MVS instance where the subsystem lives. The scheduling environment is discovered by the converter and is thus not available to JES when determining which system should be used for the conversion process.

JES

Generically, the JES role is to:

- Save the resource affinity scheduling information discovered by converter in a JES specific control block, where it can be retrieved as needed.
- Following conversion processing, invoke the WLM IWMSEDES service to determine on which systems the scheduling environment is valid and allow initiator selection of the batch job only on those systems.
- Be alert to scheduling environments becoming available or unavailable within JES's subset of the sysplex by monitoring ENF(41) and ENF(57).
- Provide inquiry support to enable the operator to determine if a scheduling environment is associated with a job, which jobs are referencing a designated scheduling environment, whether or not a job is being delayed by scheduling environment unavailability. Only JES knows which jobs have a scheduling environment associated with them.

- Provide job usage information of the scheduling environment by adding the name of the scheduling environment to the SMF26 record and passing the name of the scheduling environment to the initiator for inclusion in the SMF30 record.

A scheduling environment is an additional scheduling requirement and does not override or eliminate any current scheduling mechanism (e.g., duplicate job name, system unavailable, maximum number of jobs of given type in execution, JES managed resource unavailable in the JES3 case, etc.)

JES2 Specific Issues

Scheduling Environment Availability

Resource affinity scheduling requires that each JES2 instance in a MAS listen for ENF(57) which indicates that a previously unavailable scheduling environment has become available. Once a scheduling environment becomes available, JES2 is obligated to remove the ‘scheduling environment hold’ condition from the job and consider it for execution on the appropriate system.

Initialization

During JES2's absence, the state of resource elements could have changed resulting in scheduling environment becoming available. Each JES2 instance in a MAS must examine the jobs in the input queue having scheduling environment requirements, updating the system eligibility mask to reflect the current conditions on its MVS system.

It is possible that a job's scheduling environment no longer exists as the result of a WLM administrative action. JES2 retains the job on the input queue in a ‘scheduling environment error hold’ condition. At such time as the scheduling environment is redefined in the active WLM policy, the job will be eligible for normal scheduling.

Spool Off-load/Reload

Since jobs reloaded to spool are sent back through converter processing there are no special JES2 considerations. Resource affinities are discovered as if the job entered the system for the first time. If the scheduling environment no longer exists, the job will receive a JCL error upon reload.

Poly-JES

There are no poly-JES considerations for the use of scheduling environments.

Inquiry/Modify

JES2 provides two levels of inquiry support at the input job queue level and the individual job level.

1. The $D JOBQ command allows the operator to display

    - Jobs having a specified scheduling environment,

   - List reasons why a job is not currently eligible for execution.
   - List the systems on which a job is eligible for execution.
   - List all systems for which the scheduling environment requested by the job is available.

2. The $D J command allows the operator to display the following information for a specific job.

   - List the requested scheduling environment,
   - List reasons why a job is not currently eligible for execution.
   - List the systems on which a job is eligible for execution.
   - List all systems for which the scheduling environment requested by the job is available.

JES2 does not provide operator capability to determine which jobs are in a ‘scheduling environment error hold’ condition and thus are not allowed to be selected for execution until the WLM policy is updated.

The following figure(s) illustrate some of the JES2 display command capability in support of resource affinity scheduling.

```
          $dj7,long
JOB00007  $HASP608 JOB(TWASIKA)
$HASP608 JOB(TWASIKA)   STATUS=(AWAITING HARDCOPY),CLASS=S,
$HASP608                PRIORITY=1,SYSAFF=(ANY),HOLD=(NONE),
$HASP608                CMDAUTH=(LOCAL),OFFS=(),SECLABEL=,
$HASP608                USERID=DEALLOC,SPOOL=(VOLUMES=(SPOOL1),
$HASP608                TGS=2,PERCENT=0.3809),ARM_ELEMENT=NO,
$HASP608                SRVCLASS=HOTBATCH,SCHENV=DB2

          $djobq,schenv=DB2
JOB00007  $HASP608 JOB(TWASIKA)
$HASP608 JOB(TWASIKA)   STATUS=(AWAITING HARDCOPY),CLASS=S,
$HASP608                PRIORITY=1,SYSAFF=(ANY),HOLD=(NONE),
$HASP608                CMDAUTH=(LOCAL),OFFS=(),SECLABEL=,
$HASP608                USERID=DEALLOC,SPOOL=(VOLUMES=(SPOOL1),
$HASP608                TGS=2,PERCENT=0.3809),ARM_ELEMENT=NO,
$HASP608                SRVCLASS=HOTBATCH,SCHENV=DB2
```

There is **no** JES2 command to modify the scheduling environment assigned to the job.

Job Selection

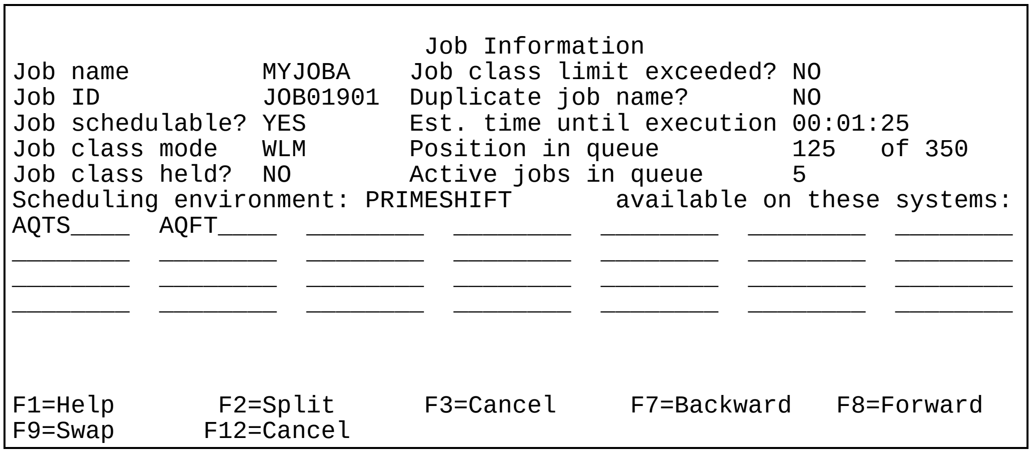
Following conversion, JES2 determines the systems on which the job's requested scheduling environment is available. During job selection processing, JES2 must verify the requested environment is available on that MVS instance. If the environment is not available, the job is marked as having a ‘scheduling environment hold’ condition and not considered for initiation on that MVS instance until WLM notifies JES2 of scheduling environment availability via ENF(57).

System Display and Search Facility

SDSF has been enhanced to provide resource affinity scheduling information to the end user and a scheduling environment control interface for system operators.

End User Support: From the ‘SDSF INPUT QUEUE DISPLAY’ or the ‘STATUS DISPLAY’ panels, a new action character causes display of resource information for the designated job in a pop-up window.

The following figure illustrates the new pop-up window.

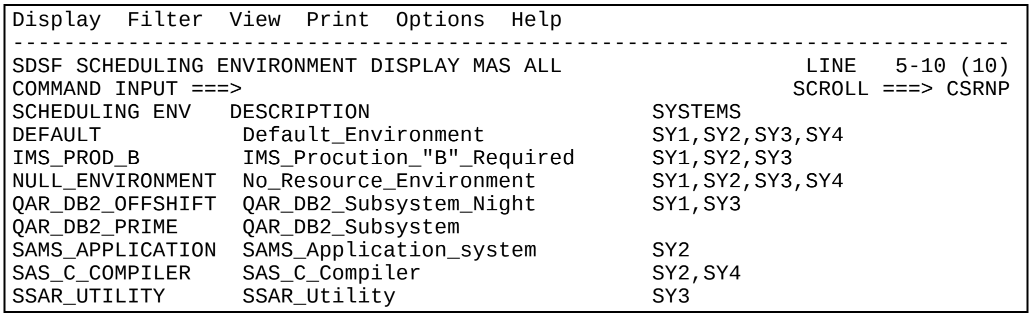


```
                            Job Information
Job name         MYJOBA    Job class limit exceeded? NO
Job ID           JOB01901  Duplicate job name?       NO
Job schedulable? YES       Est. time until execution 00:01:25
Job class mode   WLM       Position in queue         125   of 350
Job class held?  NO        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
```

Among other information, the panel displays the name of the required scheduling environment and the members of the MAS (not necessarily all the members of the sysplex) on which the scheduling environment is currently available. Information in this pop-up window is not modifiable. This display is used to determine the cause of a job waiting for execution. Other displays are available to delve into suspected scheduling environment configuration anomalies.

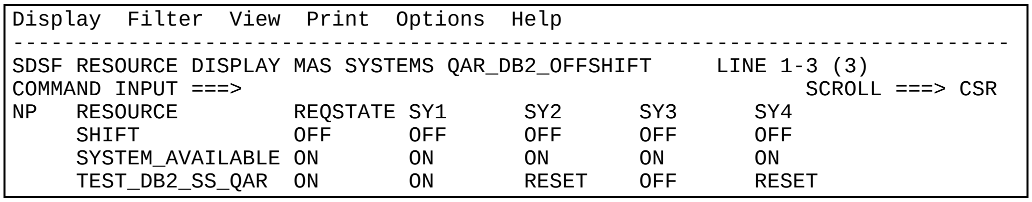
System Operator Support: There are new MAS and sysplex level panels provided to display and alter scheduling environments and resource elements on systems in the MAS or in the sysplex. Sysplex or MAS view is at the discretion of the user.

The ‘SCHEDULING ENVIRONMENT DISPLAY’ provides a table of all the scheduling environments, with the optional description and a list of all the members of the MAS on which the environment is available. The information content of this panel is the same as offered by the D,WLM,SCHENV=* command, only scoped for the JES2 MAS rather than the entire sysplex.



```
Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF SCHEDULING ENVIRONMENT DISPLAY MAS ALL                  LINE   5-10 (10)
COMMAND INPUT ===>                                          SCROLL ===> CSRNP
SCHEDULING ENV   DESCRIPTION                      SYSTEMS
DEFAULT           Default_Environment             SY1,SY2,SY3,SY4
IMS_PROD_B        IMS_Procution_"B"_Required      SY1,SY2,SY3
NULL_ENVIRONMENT  No_Resource_Environment         SY1,SY2,SY3,SY4
QAR_DB2_OFFSHIFT  QAR_DB2_Subsystem_Night         SY1,SY3
QAR_DB2_PRIME     QAR_DB2_Subsystem
SAMS_APPLICATION  SAMS_Application_system         SY2
SAS_C_COMPILER    SAS_C_Compiler                  SY2,SY4
SSAR_UTILITY      SSAR_Utility                    SY3
```

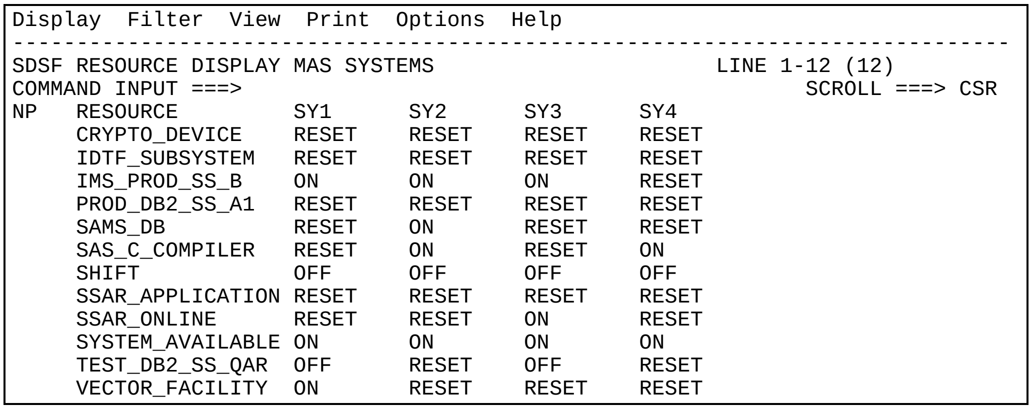
A new action character, 'R', provides detailed information on a designated scheduling environment as illustrated in the following figure.



```
Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF RESOURCE DISPLAY MAS SYSTEMS QAR_DB2_OFFSHIFT     LINE 1-3 (3)
COMMAND INPUT ===>                                           SCROLL ===> CSR
NP   RESOURCE          REQSTATE SY1      SY2      SY3      SY4
     SHIFT             OFF      OFF      OFF      OFF      OFF
     SYSTEM_AVAILABLE  ON       ON       ON       ON       ON
     TEST_DB2_SS_QAR   ON       ON       RESET    OFF      RESET
```

The details display for a specific scheduling environment lists all resource elements with their required states in the first two columns. Columns 3 - n present the state of the resource element on each member of the MAS. The state of the resource element for each MAS member may be over typed, allowing the operator to set the state of the resource element on the appropriate system..

A third panel allows a system direct MAS access to all resource elements. The following figure illustrates the new panel.



```
Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF RESOURCE DISPLAY MAS SYSTEMS                      LINE 1-12 (12)
COMMAND INPUT ===>                                           SCROLL ===> CSR
NP   RESOURCE          SY1      SY2      SY3      SY4
     CRYPTO_DEVICE     RESET    RESET    RESET    RESET
     IDTF_SUBSYSTEM    RESET    RESET    RESET    RESET
     IMS_PROD_SS_B     ON       ON       ON       RESET
     PROD_DB2_SS_A1    RESET    RESET    RESET    RESET
     SAMS_DB           RESET    ON       RESET    RESET
     SAS_C_COMPILER    RESET    ON       RESET    ON
     SHIFT             OFF      OFF      OFF      OFF
     SSAR_APPLICATION  RESET    RESET    RESET    RESET
     SSAR_ONLINE       RESET    RESET    ON       RESET
     SYSTEM_AVAILABLE  ON       ON       ON       ON
     TEST_DB2_SS_QAR   OFF      RESET    OFF      RESET
     VECTOR_FACILITY   ON       RESET    RESET    RESET
```

The state field of each resource element may be over typed, allowing the operator to set the state of a resource element on any system in the MAS.

Resource Affinity Scheduling Implementation Issues

Exploitation of resource affinity scheduling is **optional** for the installation. Resource affinity scheduling should be implemented as an installation assist and not an inconvenience. The following topics provide information to be considered when implementing resource affinity scheduling as it applies to the JES-batch environment.

Environment Requirements

At a minimum:

1. OS/390 R4 must be installed on all members of the sysplex.

2. A WLM policy defining scheduling environments and resource elements must be active.

   There is **no** requirement that the MVS instances in the sysplex be in WLM goal mode to exploit resource affinity scheduling; compatibility mode is acceptable.

Meeting these two requirements provides the resource affinity scheduling infrastructure across the sysplex which the work scheduler may utilize.

For JES2 MAS's, all members of the MAS must be at the OS/390-JES2 R4 level or higher and $ACTIVATED for resource affinity scheduling to be exploited. SDSF must be at the OS/390 R4 level if used.

Multiple JES-plexes within a Sysplex

An installation may have multiple JES-plexes of the same or different flavors within a sysplex. The environment requirements listed previously hold in this situation. There is no requirement that all JES-plexes be at the same level in the sysplex. Resource affinity scheduling for batch jobs is limited to the span of the JES.

Defining Scheduling Environments and Resource Elements

There are basically two rules that should be followed when exploiting resource affinity scheduling:

> **Rule 1**: Exploitation of resource affinity scheduling should make life simpler for the user and the operations staff.
>
> **Rule 2**: Scheduling environments should be considered permanent as they are visible to the end user and may be hard coded in JCL (for which we abhor change).

Discovering Resource Elements

Determining the various resources used by jobs in the installation is not particularly difficult. In general, a resource element should normally represent no more than one resource. In fact, you will find entities that cannot be defined and managed within the context of the function provided. If the entity cannot be used with one of two scheduling states (e.g., resources requiring exclusive use), then it should be set aside.

Once the resource elements have been discovered, one must determine if they can exist in one or two mutually exclusive scheduling states. If more than two mutually exclusive states or non-mutually exclusive states are required, then additional single state resource elements must be used to manage the resource.

For each resource element, there are two schedulable states: ON and OFF. You must determine what these values mean for the resource element in your installation.

For example, if you have a prime shift and a non-prime shift, you can use a single resource element and the two allowed states to represent these mutually exclusive conditions. Call the resource element PRIME_SHIFT and let ON denote prime shift and OFF denote non-prime shift. However, if you really have first, second and third shifts to schedule then you must use more than one resource element (probably three resource elements in this case FIRST_SHIFT, SECOND_SHIFT, and THIRD_SHIFT with only the ON state having meaning).

Remember there is a finite number of resource elements that can be defined(i.e., 999) and all resource elements will be known on all systems in the sysplex. Be conservative in defining resource elements. For example, if an application suite uses 100 different data bases, all of which must be available for the application to be available, define and use a single resource element to present the set of data bases, not 100 resource elements representing each of the data bases. The more resource elements defined the greater the potential operational impact and violation of **Rule 1**.

Grouping Resource Elements into Scheduling Environments

By the time the resource elements have been discovered and their attributes defined, one probably has a good idea of the groupings of resource elements into scheduling environments. Normally, multiple scheduling environments will be available on a MVS instance at one time. List the groups of possible coexisting scheduling environments. For each group of scheduling environments, examine the underlying resource elements to make certain there are no state conflicts. A resource element can exist in only one state on an MVS instance. So if two scheduling environments are required to be available on the same MVS instance at the same time and reference a common resource element but in differing required states, you have a problem. The WLM Resource Definition List panel provides an XREF capability which allows you to see all scheduling environments referencing a resource element.

Scheduling environments usage information should be published in much the same manner as JES job class usage information. It is important that the scheduling environments make sense and the names have lasting relevance. The names of the scheduling environment's are difficult to change once in wide use. WLM implementation of resource affinity scheduling allows changing the underlying resource usage and thus what the name means; but the name is permanent. The potential to violate **Rule 2** is great and should be carefully considered to avoid user unrest at a later time.

Controlling Scheduling Environments and Resource Elements

Under normal conditions, operations and/or automation will set the state of resource elements to reflect the time of day, availability of some resource, etc. by manipulating a small number of resource elements via operator commands or API invocations. WLM processes the commands in FIFO order propagating the state of resource elements and scheduling environments throughout the sysplex.

However, during system shutdown or dry up processing the installation may need to disable all scheduling environments on a MVS instance. If there are large number of resource elements this process can take quite some time, is error prone due to typing or requires constant maintenance of some canned command. There is an alternative solution that allows all desired scheduling environments to be disabled by changing a single resource element state.

Control resource elements

The trick to controlling all (or at least a large number of) scheduling environments is to place a control resource element in each scheduling environment. This resource element represents the availability of the scheduling environment on the system. By setting the state of the control resource element to RESET state, all scheduling environments containing the control resource element will be made unavailable on that MVS image.

Changing one resource element state is much simpler from a operator or automation point of view than changing a great number. It also causes WLM less churning than responding to several hundred modify commands.

For example, it may be worth considering defining a resource element named ‘SYSTEM_AVAILABLE’ and including it in every scheduling environment. This resource element represents the availability of the underlying MVS system. With this resource element defined in every scheduling environment (with a required state of ON), all scheduling environments on a specific system can be made unavailable through a single operator command: `F WLM,RESOURCE=SYSTEM_AVAILABLE,RESET`

COMMNDxx and IEACMD00

As previously discussed, all resource elements are set to the RESET state following IPL. It is the responsibility of operations and/or automation to set the proper state of the resource elements after IPL at the proper time. Some resources may indeed be available at IPL, such as a compiler. On first thought, one might attempt to set resource elements to the desired state through PARMLIB members COMMNDxx or IEACMD00. However, at the time the commands imbedded in these PARMLIB members are processed, WLM is not available to take direction. WLM issues message:

```
IWM041I WORKLOAD MANAGEMENT ADDRESS SPACE MODIFY COMMAND AVAILABLE
```

Following message IWM041I, the WLM modify command is available; but this occurs too late for commands in either of these members. Most likely this message will be issued prior to the automation instance initializing and thus cannot be used as a trigger either.

JES Modifications

There have been a number of resource affinity scheduling implementations by installations over the years as JES2 or JES3 extensions. External specification of required resources has been implemented through a JECL statement such as specifications.

Migration from the current JES based implementation to a the MVS provided facility requires the conversion from these JES based implementations. The migration issues are similar to those arising from moving sysout delivery information from the /*OUTPUT or //*FORMAT statement to the // OUTPUT statement. There are essentially two polar opposite approaches to this migration.

1. Discontinue the support of the JECL statement (i.e., treat statement as a comment). Require users to explicitly code a SCHENV= value on their job card to use resource affinity scheduling.

2. Convert the JECL provided resource information into a single scheduling environment in a JES input service exit, and associate the derived scheduling environment with the job following JCL conversion.

JES2 exit points

JES2 provides multiple exit points which can be used to assign a scheduling environment to a job.

- EXIT 2 - Job Card Exit.

  An existing SCHENV=scheduling_environment can be modified or deleted from the job card. A SCHENV=scheduling_environment can be added to the job card. The resulting job card is passed into the MVS converter for analysis.

  It is important to understand that information placed in the JES2 scheduling environment information area (i.e., JQASCHE) will not be over ridden by JES2 at a later time. If there is a user supplied scheduling environment on the job card, it will be ignored by JES but not the converter. Any scheduling environment specified on the job card will still be validated by converter unless removed in EXIT 6. If the scheduling environment is set in this exit, the installation must provide a valid scheduling environment or the job will not schedule into execution.

- EXIT 4 - JECL and JCL Statements Exit

  As JES is processing the input job stream, the installation may be able to determine which scheduling environment should be associated with the job and can instruct JES2 to use a specific scheduling environment by updating the JES2 data area containing the scheduling environment information.

- EXIT 4 is the logical control point to convert from installation developed, JECL based scheduling environment implementations to WLM based resource affinity scheduling.

  It is important to understand that information placed in the JES2scheduling environment information area (i.e., JQASCHE) will not be overridden by JES2 at a later time. If user supplied scheduling environment exists on the job card, the user supplied value will be ignored by JES, but not the converter. Any scheduling environment specified on the job card is still be validated by converter unless removed in EXIT 6. If the scheduling environment is set in this exit, the installation must provide a valid scheduling environment or the job will not schedule into execution.

- EXIT 20 **-** End of Reader Exit

  Upon the completion of processing the input job stream and prior to conversion, the installation may be able to determine which scheduling environment should be associated with the job and can instruct JES2 to use a specific scheduling environment by updating the JES2 data area containing the scheduling environment information.

  It is important to understand that information placed in the JES2 scheduling environment information area (i.e., JQASCHE) will not be overridden by JES2 at a later time. If user supplied scheduling environment exists on the job card, it is ignored by JES but not the converter. Any scheduling environment specified on the job card will still be validated by converter unless removed in EXIT 6. If the scheduling environment is set in this exit, the installation must provide a valid scheduling environment or the job will not schedule into execution.

- EXIT 6 **-** Converter Exit

  The installation can examine the internal text for the job card and modify/add/delete scheduling environment text unit. Any changes to the internal text will be validated after EXIT 6 returns control and jobs with invalid scheduling environment specifications will be failed with a JCL error.

  EXIT 6 can also supply a scheduling environment directly to JES2 by directly placing a value in the JES2 scheduling environment information field. Any scheduling environment specified found in the internal text will be validated by converter unless removed in EXIT 6. If the scheduling environment is set in this exit, the installation must provide a valid scheduling environment or the job will not schedule into execution.

The important thing to remember when using the JES2 exits to provide or override a scheduling environment is that JES2 does not alter a non-null value with any scheduling environment it obtains from the CI internal text following the call to exit 6.

JES Scheduling Enhancement Products

There are products on the market for JES2 that provide a resource affinity scheduling function. The functions provided by these products may or may not exceed those provided by

WLM and JES. Resource affinity scheduling function is not intended as a replacement for these products. However, the intent is to provide primitives upon which they can build.

Automatic Restart Manager

There are Automatic Restart Manager (ARM) considerations for batch jobs restarted in same-system or cross-system restart situations. ARM is intended to provide a restart function for server address spaces that presumably provide scheduling environments for other address spaces.

ARM and WLM do not have any knowledge of a job's initiator or scheduling environment requirements. If there are no initiators available then the job will wait. Similarly, if the scheduling environment is not available the job will wait.

Resource Monitoring Facility

There are no Resource Monitoring Facility (RMF) changes explicitly in support of this function. RMF does report on job queue delays by service class which reflects scheduling environment availability delays.

Service Level Reporter or Enterprise Data Manager

There is no Service Level Reporter (SLR) or Enterprise Data Manager (EPDM) support explicitly provided. There is additional information in the SMF30 records which may be used by the installation to determine scheduling environment usage and delays.