900 Series HP 3000 Computer Systems

Using the HP 3000 Workload Manager

HP Part No. B3879-90001
Printed in U.S.A. 1995

First Edition
E0195

Printing History

The following table lists the printings of this document, together with the respective release dates for each edition. The software version indicates the version of the software product at the time this document was issued. Many product releases do not require changes to the document. Therefore, do not expect a one-to-one correspondence between product releases and document editions.

Edition	Date	Software Version
First Edition	January 1995	C.50.00

Preface

This manual, *Using the HP 3000 Workload Manager*, is an introduction to using the Workload Manager on the 900 Series HP 3000 computer systems. It provides background information on traditional CPU scheduling (prior to Release 5.0), explains workgroups, and describes the procedures and processes for partitioning the system workload into workgroups.

This manual is written primarily in task format. To perform a task, simply look it up in the table of contents or in the index, turn to that section, read the introduction and follow the steps and examples. The information in Chapter Five ("Commands Reference") and the Appendices is arranged alphabetically and should be used as reference material.

In This Book

This manual is your guide to using the Workload Manager on the 900 Series HP 3000.

Chapter 1 *Introducing the Workload Manager* provides an overview of the Workload Manager and introduces workgroups.

Chapter 2 *Planning for Workgroups* provides guidelines for translating the specific needs of system users into an effective workgroup configuration. It also includes an overview of key process scheduling concepts.

Chapter 3 *Creating Workgroups* teaches the system manager how to create, modify and purge user-defined workgroups. It also describes how to replace the entire workgroup configuration now and at a scheduled time in the future.

Chapter 4 *Tuning System Performance* explains how to change the scheduling characteristics of user-defined workgroups to optimize system performance and how to use the Workload Manager to respond to performance problems as they arise.

Chapter 5 *Commands Reference* documents each of the eight commands you will use with workgroups.

Appendix A *Troubleshooting* provides a list of common problems you may encounter using the Workload Manager and suggests various solutions.

Appendix B *Error Messages* lists the messages that you can encounter when using workgroups, describes a probable cause, and suggests one or more actions to remedy the error.

In addition, a Glossary and an Index help you find, use and understand the information in this manual.

This manual contains information subject to change without notice.

Conventions

`UPPERCASE` In a syntax statement, commands and keywords are shown in uppercase characters. The characters must be entered in the order shown; however, you can enter the characters in either uppercase or lowercase. For example:

```
COMMAND
```

can be entered as any of the following:

```
command     Command     COMMAND
```

It cannot, however, be entered as:

```
comm        com_mand    comamnd
```

italics In a syntax statement or an example, a word in italics represents a parameter or argument that you must replace with the actual value. In the following example, you must replace *filename* with the name of the file:

`COMMAND` *filename*

bold italics In a syntax statement, a word in bold italics represents a parameter that you must replace with the actual value. In the following example, you must replace ***filename*** with the name of the file:

`COMMAND(`***filename***`)`

punctuation In a syntax statement, punctuation characters (other than brackets, braces, vertical bars, and ellipses) must be entered exactly as shown. In the following example, the parentheses and colon must be entered:

(*filename*):(*filename*)

underlining Within an example that contains interactive dialog, user input and user responses to prompts are indicated by underlining. In the following example, yes is the user's response to the prompt:

```
Do you want to continue? >>  yes
```

`{ }` In a syntax statement, braces enclose required elements. When several elements are stacked within braces, you must select one. In the following example, you must select either `ON` or `OFF`:

`COMMAND` { `ON` / `OFF` }

`[ ]` In a syntax statement, brackets enclose optional elements. In the following example, `OPTION` can be omitted:

`COMMAND` *filename* `[OPTION]`

When several elements are stacked within brackets, you can select one or none of the elements. In the following example, you can select `OPTION` or *parameter* or neither. The elements cannot be repeated.

`COMMAND` *filename* [`OPTION` / *parameter*]

Conventions (continued)

`[ ... ]`

In a syntax statement, horizontal ellipses enclosed in brackets indicate that you can repeatedly select the element(s) that appear within the immediately preceding pair of brackets or braces. In the example below, you can select *parameter* zero or more times. Each instance of *parameter* must be preceded by a comma:

`[,`*parameter*`][...]`

In the example below, you only use the comma as a delimiter if *parameter* is repeated; no comma is used before the first occurrence of *parameter*:

`[`*parameter*`][,...]`

`| ... |`

In a syntax statement, horizontal ellipses enclosed in vertical bars indicate that you can select more than one element within the immediately preceding pair of brackets or braces. However, each particular element can only be selected once. In the following example, you must select **`A`**, **`AB`**, **`BA`**, or **`B`**. The elements cannot be repeated.

$$\left\{ \begin{matrix} \texttt{A} \\ \texttt{B} \end{matrix} \right\} \texttt{| ... |}$$

`...`

In an example, horizontal or vertical ellipses indicate where portions of an example have been omitted.

`Δ`

In a syntax statement, the space symbol `Δ` shows a required blank. In the following example, *parameter* and *parameter* must be separated with a blank:

`(`*parameter*`)Δ(`*parameter*`)`

The symbol indicates a key on the keyboard. For example, RETURN represents the carriage return key or Shift represents the shift key.

CTRL*character*

CTRL*character* indicates a control character. For example, CTRL**`Y`** means that you press the control key and the Y key simultaneously.

Contents

Tables

1

Introducing the Workload Manager

This chapter introduces you to the Workload Manager, the new CPU monitoring and scheduling tool available for Release 5.0 of MPE/iX. It includes a discussion of the following topics:

- An overview of the features and benefits of the Workload Manager
- A review of traditional CPU scheduling
- The relationship between traditional scheduling subqueues and system-defined workgroups
- An introduction to user-defined workgroups
- A summary, in table form, of the new and modified CI commands

Introducing the Workload Manager

The Workload Manager gives system managers greater control over **CPU scheduling** at the **process** level than that provided by **traditional scheduling subqueues**. As a result, system managers can monitor and effectively manage system performance. For example, the Workload Manager allows system managers to provide a designated level of service to their customers, such as maintaining an average response time or providing a certain degree of throughput. Or, system managers can use the Workload Manager to guarantee a minimum percentage of the CPU to a set of users, or restrict them to a maximum percentage of the CPU.

With this release of MPE/iX, the traditional scheduling subqueues (AS, BS, CS, DS, and ES) are replaced by five **system-defined workgroups**. In addition, system managers can create additional **user-defined workgroups** for more discrete partitioning of the system workload. Specifically, you use the Workload Manager to determine:

- the total number of **workgroups** on the system
- which processes become members of each workgroup
- the behavior of processes in each workgroup, such as their priority, how much time a process can have the CPU before it's interrupted, and so on
- the current workgroup configuration

You may partition the system into as many workgroups as needed to control the workload. The workgroups can correspond to one department within the company, such as Finance, or to a group of people performing the same kind of task, such as all data entry personnel updating one database. And, since you can observe the behavior of individual workgroups, the Workload Manager enables you to closely monitor system performance.

You can also implement a new workgroup configuration on the fly or at a predetermined time, for example, just before a shift change or to handle scheduled batch jobs. In this way, you can plan for anticipated workload changes and modify system behavior accordingly.

Note

There is full programmatic support for the Workload Manager integrated into MPE/iX. For information about the Architected Interface calls available to you, consult the *MPE/iX Architected Interface Facility: Operating System Reference Manual*, part number 36374-90001.

Traditional CPU Scheduling

In traditional CPU scheduling (prior to MPE/iX Release 5.0), system and user processes run in one of five scheduling subqueues: the AS subqueue, BS subqueue, CS subqueue, DS subqueue and ES subqueue. In this scheme, scheduling subqueue is both an *entity* and a *process attribute*.

The scheduling subqueue as an entity has defined characteristics (such as base and limit, the minimum and maximum quantum, timeslice, and the boost property) that determine the scheduling policies of processes running in that subqueue. You can think of the scheduling subqueue as a dynamically changing collection of processes. The scheduling subqueue is a process attribute when, for example, a job is described as "having a priority of CS" or, for example, when "a program is running at DM priority".

Users with system supervisor (OP) or system manager (SM) capability can affect the behavior of the scheduling subqueues as entities. Such users may alter, or **tune**, the characteristics of the CS, DS and ES scheduling subqueues. (These subqueues, which collect user processes, are also known as the **circular scheduling subqueues**). No users, regardless of capability, can alter the characteristics of the AS and BS subqueues, which are reserved for system processes.

Users without special capabilities can influence the scheduling subqueue as a process attribute through one of the following means:

- Users can choose a scheduling subqueue via the `;PRI=` option of the `HELLO`, `RUN` or `JOB` commands. Such a change lasts for the duration of the session, program, or job. Furthermore, the system manager constrains the priority that a user can assign to a process via the `MAXPRI` option of the `NEWUSER`, `ALTUSER`, `NEWACCT`, or `ALTACCT` commands.
- Users can change the scheduling subqueue of a process via the `;PRI=` option of the `ALTPROC` command. Such a change lasts until the process terminates or until the queue attribute is changed again.
- Programmers (and system managers) can use one of two Link Editor options to determine a program's scheduling subqueue when they are creating an executable program file: the *priority_level*, to specify the default execution priority at run time or *max_priority_level*, to specify the maximum priority a program can have at run time. Within this limit, users can alter the program's priority by issuing the `;PRI=` option of the `RUN` command.
- Programmers can use either the `AIFPROCPUT` routine or the `GETPRIORITY` intrinsic to change the scheduling subqueue of a process.

Introducing Workgroups

You can think of a workgroup, like its predecessor the scheduling subqueue, as a dynamically changing collection of processes. A workgroup consists of three components: the workgroup name, membership criteria, and specific scheduling characteristics.

There are two types of workgroups which you can have on your system:

- System-defined workgroups, which come with your system and cannot be deleted.
- User-defined workgroups, which you create, modify and purge as needed to meet the changing demands of your system workload.

Each type of workgroup is explained in the following sections.

System-defined workgroups

All customers who update to version C.50.00 of MPE/iX will have five system-defined workgroups on their systems. Each system-defined workgroup corresponds to one of the five traditional scheduling subqueues. The AS subqueue, for example, becomes the AS_Default workgroup. Other than the difference in nomenclature, however, there is nothing that distinguishes the system-defined workgroup from their counterpart, the scheduling subqueue. The set of scheduling characteristics that you can control including the *base* and *limit* priorities, *min* and *max* quantum values, boost property (`DECAY` or `OSCILLATE`), and *timeslice*, as well as their behavior, remains unchanged.

Table 1-1.
A Comparison of Scheduling Subqueues and System-Defined Workgroups

Queue	Corresponding Workgroup	Membership Criteria	Scheduling Characteristics
AS	AS_Default	`MEMB_QUEUE=(AS)`	Uses default values
BS	BS_Default	`MEMB_QUEUE=(BS)`	Uses default values
CS	CS_Default	`MEMB_QUEUE=(CS)`	Uses default values; allows system manager to make changes.
DS	DS_Default	`MEMB_QUEUE=(DS)`	Uses default values; allows system manager to make changes.
ES	ES_Default	`MEMB_QUEUE=(ES)`	Uses default values; allows system manager to make changes.

Each of the system-defined workgroups has one and only one criterion for membership: the queue attribute which specifies its scheduling subqueue. This assures complete backward compatibility with previous versions of the operating system. All user processes will fall into one of these system-defined workgroups using the customary scheduling parameters and all existing programs and job scripts will continue to work.

User-defined workgroups

Using the Workload Manager, system managers can create an essentially unlimited number of user-defined workgroups to partition their system's workload. Such workgroups consist of a workgroup name, membership criteria, and scheduling characteristics. Continue reading here for a general introduction to workgroups. For complete information about creating workgroups, modifying workgroups, and changing the workgroup configuration, read Chapter 3.

Workgroup name

The workgroup name uniquely identifies the workgroup. Typically, it indicates the kinds of processes which become members of the workgroup, such as `Program_Development` or `Payroll_Batch`.

Membership in a user-defined workgroup

Membership criteria for a user-defined workgroup can be any combination of the following, but at least one criterion is required:

- Logon identity, of the form *job/session name,user.account*. The job or session name is optional, but the user and account names are required. To capture a group of processes which share a similar identity, the logon identity may include wildcard characters.
- Program name. As in logon name, the use of wildcards is permitted in the program name specification.
- Queue attribute, which is one of AS, BS, CS, DS, or ES. Note that the *queue attribute* is not the same as the *scheduling subqueue*. The attribute is one element used to determine membership in a workgroup whereas the traditional scheduling subqueue, as an entity, is comparable to the workgroup.

Each process is assigned to its appropriate workgroup at process creation, whenever a process attribute on which workgroup membership is based is changed, or after it is explicitly assigned to a workgroup.

Any command, intrinsic, or AIF that sets or changes the queue attribute of a process may also set or change the workgroup of that process (if `QUEUE` is specified in the membership criteria.) Subqueue assignment of a process can be set by `RUN`, `HELLO`, `JOB`, and `LINK`, and it can also be changed with `ALTPROC`, `AIFPROCPUT`, and `GETPRIORITY`.

Scheduling characteristics of user-defined workgroups

The scheduling characteristics the system manager can assign to a workgroup include the following:

- base and limit priorities
- quantum bounds
- boost property
- timeslice
- minimum and maximum CPU percentage bounds

These characteristics, with the exception of minimum and maximum CPU percentage bounds, should be familiar to system managers since they are characteristics of the traditional scheduling subqueues set with the `TUNE` command. CPU percentage bounds let you either guarantee that processes in a workgroup will get *at least* a specified amount of CPU time (the minimum CPU percentage) or place an upper limit on the amount of CPU that processes in a workgroup get (the maximum CPU percentage).

The next chapter, "Planning for Workgroups", explains each of these characteristics for individuals who want to review the information conceptually. Chapter 3, "Creating Workgroups", has examples of how to set and change scheduling characteristics, and all command parameters are fully documented in Chapter 5, "Commands Reference".

New and Modified Commands

The following table summarizes the new and modified CI commands that are available with Release 5.0 of MPE/iX. For a complete description of the commands, refer to Chapter 5.

Table 1-2.
Workload Manager and CPU Scheduling Commands

Command	Behavior
`ALTPROC`	This command continues to allow process attributes to be set. Use the new `;WG=` option to explicitly place a process in a workgroup. Such a process becomes an **artificial member** of the workgroup. You also use the `;WG=` option to allow a process to migrate to its natural workgroup. Note that you should **not** use the ;PRI= option to place a process at a fixed priority. Instead, create a workgroup with the base and limit set to the desired priority value and use the `;WG=` option to place the process in that workgroup.

Table 1-2.
Workload Manager and CPU Scheduling Commands
(continued)

Command	Behavior
ALTWG	Use this new command to alter any of the scheduling characteristics of an existing workgroup, including base and limit priorities, quantum bounds, boost property, timeslice, and CPU percentages. The processes belonging to that workgroup will be scheduled in accordance with the new parameters. (To alter the name of a workgroup or its membership criteria, you must use the NEWWG command.)
NEWWG	Use this new command to create a new, user-defined workgroup to add to the current configuration. You can also use NEWWG to replace the current configuration with a new one. To do so, you specify an indirect file that contains the specification for the complete set of workgroups you want to use instead.
PURGEWG	Use this command to purge the workgroup(s) specified on the command line. You may use wildcards to purge multiple workgroups and choose from a variety of prompting and display options. You also use PURGEWG to execute deferred purgescans.
SHOWPROC	This command continues to display process attributes. A new format, DETAIL, provides information regarding the attributes that can determine workgroup membership (logon, program, subqueue) and the resulting workgroup. The DETAIL display format also distinguishes processes that are artificial members of a workgroup from natural members of the workgroup.
SHOWQ	Use this command to display scheduling data for all processes and the scheduling characteristics of the CS_Default, DS_Default and ES_Default workgroups. This command has not been changed and the three system-defined workgroups appear as traditional scheduling subqueues in the left-most column of the STATUS display.
SHOWWG	Use this new command to display scheduling and process data for the specified workgroup(s). The SHOWWG command supports a variety of formats including summary information on the workgroups, member processes, and detailed information. It also offers an output format that, when redirected to a file, is suitable for input to the NEWWG command.
TUNE	You can continue to use this command to change the scheduling characteristics of the CS subqueue (CS_Default workgroup), the DS subqueue (DS_Default workgroup, and the ES subqueue (ES_Default workgroup). Or, you can use the ALTWG command to do so. You must use ALTWG to modify the scheduling characteristics of all user-defined workgroups.

2

Planning for Workgroups

This chapter gives the system manager guidelines for evaluating current system use and performance, and suggests how to translate this information into an effective workgroup configuration. It includes the following topics:

- Evaluating current system use
- Translating current usage into a workgroup configuration
- Understanding the importance of workgroup order
- Using GlancePlus to gather performance information
- Choosing appropriate scheduling characteristics

This chapter also provides an overview of the key concepts that you must understand to tune system performance. In particular, it reviews the function and use of base and limit priorities, quantum bounds, the boost property, timeslice, and CPU percentage bounds.

Planning Workgroups

To use the Workload Manager effectively, you need to know who uses your system now, how they use it, and how to translate this information into an effective workgroup configuration. In this section, you'll find questions and suggestions to help you gather this data.

To evaluate current system use

Knowledge of the current system workload is essential to the effective use of the Workload Manager. As system manager, you should be familiar with the demands of all your users and clearly understand the relative priorities of these demands.

The questions below can help you gain a more complete understanding of who uses the system and what their needs are. Consider these questions as a guideline only. Other questions, more pertinent to your particular situation, will arise while you are gathering information.

- What kinds of on-line users need the system? *Begin by categorizing users by job title or function, for example, data entry clerks, customer service representatives, system analysts, or programmers.*
- What system needs does each group of users have? *For example, do some of the users interact directly with customers and therefore require fast response time?*
- Do you have a Service Level Agreement (SLA) with a specific department or division within your company that you must meet?
- Is there an identifiable pattern of system use over time for each group? *For example, at the end of each quarter, do your financial analysts place a heavy demand on the system to prepare fiscal reports? Or is there a peak demand for database access between 9 o'clock and noon and again between 2 and 5 o'clock?*
- How and when are batch jobs submitted on your system and do any of them require a guaranteed turnaround time? *For example, does your Marketing staff run a report every morning that must be on the CEO's desk by 10 a.m.?*

To partition the system workload into workgroups

Once you understand who is using the system and what their needs are, you can determine how to group the users effectively. For example, you might create separate groups for data entry clerks, all batch jobs updating a certain database, one particular batch job, application developers using editing programs, or special servers. It is these logical groupings that you translate into workgroups.

Suppose you wanted to begin by creating three user-defined workgroups, one for a server, one for data entry clerks who log orders for retail products, and one for programmers who are editing their application files. The workgroup membership criteria might look like this:

Table 2-1. A Set of Sample Workgroups

Description	Workgroup Name	Membership Criteria
Server workgroup	Server	`MEMB_PROGRAM=(SERV.APPL.DEV)`
Order entry workgroup	Retail_online	`MEMB_LOGON=(@.ORDERS)`
Program editing workgroup	Prog_Develop	`MEMB_PROGRAM=(EDITOR.PUB.SYS, QEDIT.@.@, HPEDIT.@.@)` and `MEMB_LOGON=(@.TEST, "TST,@.DEV")`

Once you decide what workgroups you want, the next step is to determine the appropriate workgroup order and what kind of access to the CPU each workgroup will have (explained next). It is important to understand, however, that the first workgroup configuration you devise is unlikely to be the last. Developing workgroups with the appropriate membership criteria and scheduling characteristics is an iterative process. You should anticipate continual monitoring of the existing workgroup configuration, evaluation of system performance, and at least some ongoing modification.

To understand the importance of workgroup order

The workgroups on your system are maintained in an ordered list, which is identical to the order in which workgroups are listed in any indirect file used by the `NEWWG` command. You determine workgroup order by:

- Creating and/or editing the indirect file.
- Adding new workgroups to the current configuration.
- Removing a workgroup from the current configuration.

When the Workload Manager assigns a process to a workgroup, it uses a "first-fit algorithm". That is, the Workload Manager searches the list of workgroups on your system and places the process into the first workgroup in which it fits. It does not seek a "best fit".

Both the membership criteria that you do specify, and those that you don't specify, affect the placement of processes in a workgroup. That's because if you omit a category, the Workload Manager treats it as a match when it is determining membership. (For example, to allow a process with any logon potentially to become a member of a workgroup, don't specify `MEMB_LOGON=` when you create the workgroup.) In general, to capture a broader population of processes, specify fewer membership criteria.

In light of these factors, it is important that you order workgroups so that those with the most specific membership criteria appear first and those with more general membership criteria follow. The five system-defined workgroups, which only specify a single criterion, `MEMB_QUEUE`, always appear last in the list of workgroups.

How workgroup membership is determined: an example

Listed below are the names and membership criteria for three user-defined workgroups and the five default system-defined workgroups. (These are not intended to be realistic or complete workgroup specifications, since the point of this example is to understand workgroup membership.)

```
WORKGROUP=Program_Development
;MEMB_PROGRAM=(EDITOR.PUB.SYS, QEDIT.@.@, HPEDIT.@.@)
;MEMB_LOGON=(@.TEST, "NM@,@.MYTEST")

WORKGROUP=Payroll_Online
;MEMB_PROGRAM=(PAYROLL.@.PRAPP)
;MEMB_QUEUE=(CS)

WORKGROUP=Payroll_Batch
;MEMB_PROGRAM=(PAYROLL.@.PRAPP)
;MEMB_QUEUE=(DS,ES)

WORKGROUP=AS_Default
;MEMB_QUEUE=(AS)

WORKGROUP=BS_Default
;MEMB_QUEUE=(BS)

WORKGROUP=CS_Default
;MEMB_QUEUE=(CS)

WORKGROUP=DS_Default
;MEMB_QUEUE=(DS)

WORKGROUP=ES_Default
;MEMB_QUEUE=(ES)
```

Based on the preceding membership specifications, processes are assigned to the workgroups as shown in the following table.

Process Attributes				Resulting Workgroup
PIN	Logon	Queue	Program File	
100	CMTEST, CHUCK.TEST	CS	EDITOR.PUB.SYS	Program_Development
200	NMTEST, DOUG.MYTEST	CS	EDITOR.PUB.MYTEST	CS_Default
300	NMTEST, SLC.TEST	BS	EDITOR.PUB.SYS	Program_Development
400	CMTEST, DOUG.MYTEST	BS	HPEDIT.PUB.SYS	BS_Default
500	NMTEST, SLC.MYTEST	BS	QEDIT.PUB.SYS	Program_Development
600	SUSAN, SLC.PRAPP	CS	PAYROLL.PUB.PRAPP	Payroll_Online
700	JREPORT, GREG.MYTEST	DS	PAYROLL.RPT.PRAPP	Payroll_Batch

To help you understand this example, consider why process 100 is assigned to the `Program_Development` workgroup and process 200 to the `CS_Default workgroup`. Both processes have the same queue attribute, and both meet the logon requirement for `Program_Development`. The critical difference is that process 200 is executing `EDITOR` from the `MYTEST` account, whereas the `Program_Development` workgroup requires that `EDITOR` come from `PUB.SYS`.

To find the workgroup for which Process 200 does qualify, proceed down the ordered list. Process 200 fails to match the program criterion for `Payroll_Online` and the queue criterion for `Payroll_Batch`, `AS_Default`, and `BS_Default`. The first workgroup in which Process 200 does fit is `CS_Default`, and, as a result, that is its destination.

To establish workgroup scheduling characteristics

The workgroup membership criteria determine which processes are grouped together in a particular workgroup, while the scheduling characteristics control the CPU access of those processes. You use your knowledge of the needs and relative priorities of these workgroups to establish the scheduling characteristics.

When you first create workgroups, consider setting only the required scheduling characteristics of base and limit priority, allowing all other values to assume their defaults. Then, monitor system performance and make the necessary adjustments, using the following list of guidelines to help you.

- Give higher base and limit priorities to more important workgroups.
- Assign overlapping priority ranges to workgroups that have similar needs.
- Adjust quantum bounds to provide the proper rate of priority decay.
- To allow workgroups with long-running transactions to compete with shorter transactions, do one of the following: set the boost property to `OSCILLATE`, increase the timeslice value, or increase the minimum CPU percentage of the workgroup.

If you need to review the function and use of any of the traditional scheduling parameters, read the next section. It explains each of them in detail, as well as describing the new CPU percentage bounds.

An Overview of Scheduling Characteristics

With Release 5.0, the MPE/iX Dispatcher remains priority-driven, dispatching processes to the CPU(s) based on their priority. The MPE/iX Scheduler controls process priorities in accordance with the scheduling parameters established by the system manager. The scheduling characteristics include those characteristics associated with traditional scheduling subqueues, as well as new CPU percentage bounds that are available for user-defined workgroups.

Base and Limit Priorities

The base and limit priorities determine the range of priorities available to processes within the workgroup. If no user-defined workgroups have CPU percentage minimums, the CPU will be allocated to processes based on their priorities. For example, processes in a workgroup with base=152 and limit=160 will run before processes in a workgroup with base=170 and limit=180. (Note, however, that the priorities you assign to workgroups can overlap, so that processes within the workgroups compete with each other for the CPU.)

For base and limit, priority is inversely related to the integer value: a higher-priority process has a lower number. Processes in the CS_Default, DS_Default, and ES_Default system-defined workgroups, and all processes in any user-defined workgroups, begin execution at the base priority and decay toward the limit priority. At some point, depending upon the boost property set for the workgroup, any process that is not yet complete is boosted back to the base priority to continue execution.

In traditional MPE/iX parlance, the CS, DS, and ES scheduling subqueues were known as “circular scheduling queues” because their process’ priority circulated within the bounds of the established base and limit values for the subqueue. For example, a process in the CS subqueue will start at the base priority of 152 and decay towards the limit of 200. This behavior has not changed for the three system-defined workgroups which corresponding to the CS, DS, and ES subqueues.

Note

The introduction of workgroups dictated a simplification of the preemption algorithm. Because the assumption that a higher subqueue indicates a more important process is no longer valid, the dependency on scheduling queue attribute was eliminated. The algorithm continues to compare priorities, only considering preemption if the priority of the potential preemptor exceeds the priority of the running process by the preemption threshold. If system and boosted processes are able to pass these priority checks, they can preempt the running process (if it is preemptable).

Quantum Bounds

The minimum and maximum quantum values bound the calculation of the workgroup quantum, which determines the rate of priority decay of processes within the workgroup. The quantum represents the average transaction time of processes within that workgroup. The consumption of CPU by a process is compared against the quantum to determine the amount of priority decay. Small quantum values mean most transactions are short, and process priorities will decay quickly. Larger quantum values indicate longer transaction times, and process priorities will decay more slowly.

Note

All three system-defined workgroups that have decayable processes, the CS_Default, DS_Default and ES_Default workgroups, now perform a dynamic quantum calculation using a similar decay algorithm. The algorithm bases priority decay on the quantum divided by a constant, applying smaller amounts of priority decay for smaller amounts of CPU consumption.

The default values for the minimum and maximum quantum for the DS_Default and ES_Default workgroups have remained the same to ensure that the quantum will not vary unless you choose to change the bounds. The traditional DS and ES subqueues did not provide this flexibility and had a fixed quantum value.

Boost Property

While the quantum controls the rate of priority decay, the boost property determines the behavior of the process once its priority has decayed to the limit of the workgroup. The default value of `DECAY` indicates that the process will decay to the limit and remain there until it completes its transaction. The value `OSCILLATE` indicates that if the process priority decays to the limit of the workgroup, its priority is reset to the base priority (the process oscillates between the base and limit priorities).

Timeslice

The timeslice is used to ensure that one process does not monopolize the CPU for long periods of time. When a process is launched, the Dispatcher guarantees that it will not run for more than its timeslice value (even if it is CPU-bound). The Dispatcher will actually take the CPU away from the process if it is still running after the timeslice interval has passed (provided that the process can be interrupted).

Note

Before Release 5.0 of MPE/iX, CS subqueue processes were able to preempt DS or ES subqueue processes since they had a higher subqueue, and thus the timeslice values for the DS and ES subqueue were set high. These default values have now been reduced so that processes in these workgroups cannot monopolize the CPU for long periods of time. The timeslice value remains tuneable, however, and the system manager can adjust it to a higher value if it is appropriate to the environment.

CPU Percentage Bounds

User-defined workgroups also allow the specification of minimum and maximum CPU percentage bounds. The minimum percentage serves as a guarantee. Processes in a workgroup with minimum CPU percentage equal to 20% will be guaranteed 20% of the CPU(s), provided they have enough demand to use the 20%. If the processes demand more than 20%, they can receive more, providing it does not violate the minimum values for other workgroups. Thus, the minimum is a true minimum and can be exceeded. It is not a target amount.

The maximum percentage serves to restrict the CPU consumption of a workgroup. Processes in a workgroup with a maximum CPU percentage equal to 50% will **never** receive more than 50% of the CPU. If no other workgroups require CPU, the system will idle rather than allow the workgroup to exceed its maximum.

The sum of all CPU minimums should not exceed the amount necessary to provide sufficient "leftover" CPU for the system processes. As an alternative to guaranteeing a minimum to workgroups, the system manager might choose to set a maximum CPU percentage on a workgroup that tends to starve other workgroups. The maximum will constrain the workgroup to run within the allocated amount of CPU.

Gathering Data for the Workload Manager

In the following sections, you'll learn some techniques for using HP GlancePlus with the Workload Manager. You can use the performance tool of your choice to evaluate current system performance and, once you have set up workgroups on your system, evaluate their behavior.

Using GlancePlus to gather information

To use Workload Manager with GlancePlus, you edit the configuration file `Glancnfg` to add workgroups to an existing application class specification or to define a new application class that corresponds to a workgroup. When you associate the new `WORKGROUP` keyword with a Workload Manager workgroup, the resources used by this process will be added to the resources used by other processes in the application class.

You can set up GlancePlus applications in three ways, each of which is explained below.

- One application class to one workgroup.
- One application class to many workgroups.
- One workgroup to many application classes.

Assigning one application class to one workgroup

To define an application class that maps exactly to a single workgroup, you enter the application name and the name of the workgroup in the configuration file. For example, the following specification defines an application called `Data Entry` and maps it to the `Data_Entry` workgroup:

```
APPLICATION=Data Entry
WORKGROUP=Data_Entry
```

This application class contains all processes that are members of the `Data_Entry` workgroup, and no other processes.

Assigning one application class to many workgroups

You may want to define an application class that partitions the processes in a workgroup into discrete groups. For example, suppose you have defined the following `Sales` workgroup:

```
:newwg Sales;memb_logon=("dept@.@.sales");base=160;limit=180
```

You can then define two different applications to view the processes in this one workgroup by adding this application class specification to the `glancnfg` file:

```
APPLICATION=Dept1
USER="dept1@.@.sales"

APPLICATION=Dept2
USER="dept2@,@.sales"
```

Assigning one workgroup to many application classes

There may be a situation in which you want one application class to provide data on multiple workgroups. For example, suppose that you have defined two workgroups on your system, `Billing` and `Payroll`, but for the purpose of performance evaluation, you can lump them into one application called `Finance`. To do so, you would add this application class specification to your `glancnfg` file:

```
APPLICATION=Finance
WORKGROUP=Billing, Payroll
```

Viewing workgroup data in GlancePlus

Once you have set up workgroups in the `Glancnfg` file, you can use the new Workload Manager screen to display information about workgroups. To access the Workload Manager screen, issue the `K` command from any window or, at the GlancePlus global window, press the Workload Manager softkey `F1`.

Note

To analyze a portion of the system workload, use the Filter screen to select the information that you want to see. Press the `F` key to define a filter, enter the appropriate information, (program, user, or application), and press `Return`.

3

Creating Workgroups

This chapter teaches you how to create user-defined workgroups for your system. It includes the following topics:

- Creating workgroups with the `NEWWG` command
- Making a copy of the current workgroup configuration
- Replacing the current configuration with a new set of workgroups
- Displaying the current workgroup configuration with the `SHOWWG` command
- Modifying a workgroup's scheduling characteristics with the `ALTWG` command
- Purging a workgroup with the `PURGEWG` command

Adding New Workgroups

Using Workload Manger, you can create an essentially unlimited number of user-defined workgroups for your system. When you create a workgroup, you specify three kinds of information: the workgroup name, membership criteria, and scheduling characteristics.

You can use the NEWWG command to add a new workgroup to the current configuration or to replace the entire workgroup configuration with a new one. For instructions to replace the workgroup configuration, read "Creating a New Workgroup Configuration" later in this chapter.

To create a new workgroup

To add a workgroup to the existing configuration, use the NEWWG command to specify the name, one or more membership criteria, and the scheduling characteristics. The syntax of the NEWWG command is:

```
NEWWG [WORKGROUP=]workgrp
{ [;MEMB_LOGON=]logon          }
{ [;MEMB_PROGRAM=]program_file }
{ [;MEMB_QUEUE=]queue_attribute }
[;BASE=]base  [;LIMIT=]limit
[[;MINQUANT=]min]  [[;MAXQUANT=]max]
[ [;BOOST=]{ DECAY     } ]
[          { OSCILLATE } ]
[[;TIMESLICE=]tslice]
[[;MINCPUPCT=]minpercent]  [[;MAXCPUPCT=]maxpercent]
[[;POSITION=]existingwg]
```

For more detailed information on this command, refer to Chapter 5, "Commands Reference".

Note

To replace the existing workgroup configuration by specifying an indirect file on the NEWWG command line, you use a different syntax. For information, see "Creating a New Workgroup Configuration" in this chapter.

For example, to create a new user-defined workgroup named Program_Development whose base and limit priorities are 160 and 170 (respectively) and with a boost property of OSCILLATE, enter:

```
:NEWWG Program_Development; memb_program= (editor.pub.sys,&
qedit.@.@, hpedit.@.@);memb_logon=("nm@,@.mytest"); base=&
160; limit= 170; boost= oscillate
```

Or, using positional parameters instead of keywords, you would enter:

```
:NEWWG Program_Development, "nm@,@.mytest", (editor.&
pub.sys,qedit.@.@,hpedit.@.@),,160,170,,,oscillate
```

The membership criteria must appear in the order MEMB_LOGON, MEMB_PROGRAM and MEMB_QUEUE when the keywords are not specified and, in this case, since MEMB_QUEUE isn't specified, a comma is used as a placeholder. The ampersand character (&) appears at the end of the command lines in the examples above to indicate continuation. If you type the command on one single line (so that it wraps), omit the ampersand.

For a process to become a natural member of the Program_Development workgroup (that is, one that naturally meets the membership criteria), both of the following conditions must be true:

- The user (or job) must log onto any user name in the account mytest, with a job or session name that begins with nm. For example, a user logging on as nmbuild1,pat.mytest qualifies. Or, a job identified as nmbeta,doug.mytest qualifies.
- The user or job logging on must run one of three editing programs editor.pub.sys, qedit.@.@, or hpedit.@.@.

To place the workgroup in the ordered list

The workgroups on your system are maintained in an ordered list. When the Workload Manager assigns a process to a workgroup, it searches the list of workgroups on your system and places the process into the **first** workgroup in which it fits. Since workgroup membership dictates how a process is scheduled, which workgroup it falls into is critical.

When you create new workgroups, you can use the POSITION parameter to determine where the workgroup is positioned in the ordered list. You should order workgroups so that those with the most specific membership criteria appear first and those with more general membership criteria follow. For example, to create a workgroup named Payroll_online for online users of the Payroll account that appears ahead of the Program_Development workgroup (created in the preceding example), enter:

```
:NEWWG Payroll_online; memb_logon= (@.Payroll); &
base= 170; limit= 190; position= Program_Development
```

Or, to use positional parameters, you enter the command this way:

```
:NEWWG Payroll_online, @.Payroll,,,170,190,,,,,,,&
Program_Development
```

Note that you must use the comma as a placeholder for the membership criteria you are not specifying. The ampersand character (&) appears at the end of the command lines in the examples above to indicate continuation.

For a process to become a natural member of the `Payroll_online` workgroup, the user must logon to the `Payroll` account with any user name. (The user can also choose a session name. Since it isn't specified in the membership criteria, any session name is considered a match.) The `Payroll_online` workgroup has a base priority of 170, a limit of 190, the default minimum and maximum quantum of 1 and 2000 milliseconds, the default boost property (`DECAY`), and the default timeslice of 200 milliseconds.

For more conceptual information and an example of determining workgroup membership, read "To understand the importance of workgroup order" and "How workgroup membership is determined: an example", in Chapter 2.

To display the workgroup

To display information about all workgroups on the system, you use the `SHOWWG` command. You can choose to display information in any of the formats available: `SUMMARY`, `DETAIL`, `PROCS`, or `WGFILE`. If you do not specify a format (as shown below), the `SHOWWG` command defaults to the `SUMMARY` format.

For example, assuming that you had just created the two workgroups shown in the previous examples, entering the `SHOWWG` command would produce this information:

```
:SHOWWG


                                -----QUANTUM-----          TIME          CPU %
   WORKGROUP          BASE LIMIT   MIN    MAX  ACTUAL  BOOST  SLICE PROCS  MIN MAX
------------------- ---- -----  ----- ----- ------  -----  ----- -----  --- ---
Payroll_online      170  190    1     2000  0       DECAY  200   0      0   100
Program_Development160  170    1     2000  0       OSC    200   0      0   100
AS_Default          30   99     N/A   N/A   N/A     N/A    1000  14     N/A N/A
BS_Default          100  150    N/A   N/A   N/A     N/A    1000  30     N/A N/A
CS_Default          152  200    1     2000  4       DECAY  200   21     N/A N/A
DS_Default          202  238    2000  2000  0       DECAY  200   0      N/A N/A
ES_Default          240  253    2000  2000  0       DECAY  200   0      N/A N/A
```

Creating a New Workgroup Configuration

As you have seen, you use the NEWWG command to add a new workgroup to the current configuration. You can also use it to replace the existing set of workgroups with a new set. Substituting one workgroup configuration for another is useful when you can anticipate typical changes in the workload. For example, your system may run more efficiently if you introduce a new workgroup configuration at 6 p.m. on Friday afternoons (to accommodate the typical weekend workload) and then return to a "weekday configuration" at 9 a.m. Monday mornings. Or, for example, you may need a workgroup configuration that allocates more of the CPU to the finance department at the end of each month or each quarter, allowing them to produce critical reports in a timely manner.

Replacing the current configuration requires that you perform some or all of the steps listed below. What is required in your specific situation depends upon your starting point and whether or not you are creating a new configuration (using the indirect file) for the first time.

1. Keep a copy of the current workgroup configuration as an ASCII file.
2. Edit the file you just created so that it contains the new workgroups.
3. Validate the contents of the indirect file so that the replacement will be successful.
4. Introduce the new workgroup configuration now or at a specified time in the future.

Each of these steps is described in the next five sections.

To keep a copy of the current workgroup configuration

You can use the SHOWWG command with the WGFILE format to produce a complete description of the workgroups on your system. By redirecting the command output to a file (using CI I/O redirection), you can easily create an ASCII file that is suitable as input with the NEWWG command.

To keep a copy of the current configuration, follow these steps:

1. At the MPE/iX prompt, enter:

   SHOWWG @;FORMAT=WGFILE > *filename*

   For *filename*, enter the name you want to use for the configuration file. For example, to create an indirect file named currwgs, enter:

   SHOWWG @;FORMAT=WGFILE > currwgs

   If you want to use a name that's longer than 8 characters, make sure you precede it with ./ to indicate HFS syntax.

2. To save the file you just created as a permanent file, enter:

   SAVE *filename*

You can edit the file so that it defines a new workgroup configuration. Or, you can keep it as a "backup copy" of the current configuration. Either way, you introduce these workgroups to the system by specifying the file on the NEWWG command line.

Note

The output that you get with the SHOWWG @;FORMAT=WGFILE command is a snapshot of the current system configuration. The true system state is kept internally, in tables, and not in a user file.

To edit the configuration file

The file that you produce using the WGFILE format of the SHOWWG command will resemble the sample file shown on this page and the next.

```
Workgroup     = Payroll_online
;Memb_Logon   = @.Payroll
;Base         = 170
;Limit        = 190
;MinQuant     = 1
;MaxQuant     = 2000
;Boost        = DECAY
;Timeslice    = 200
;MinCPUPCT    = 0
;MaxCPUPCT    = 100

************************
Workgroup     = Program_Development
;Memb_Logon   = "nm@,@.mytest"
;Memb_Program = editor.pub.sys,quedit.@.@,hpedit.@.@
;Base         = 160
;Limit        = 170
;MinQuant     = 1
;MaxQuant     = 2000
;Boost        = OSCILLATE
;Timeslice    = 200
;MinCPUPCT    = 0
;MaxCPUPCT    = 100

************************
COMMENT  The following are system-defined (default) workgroups.
COMMENT
COMMENT  Workgroup     = AS_Default
COMMENT  ;Memb_Queue   = AS
COMMENT  ;Base         = 13
COMMENT  ;Limit        = 99
COMMENT  ;MinQuant     = N/A
COMMENT  ;MaxQuant     = N/A
COMMENT  ;Boost        = N/A
COMMENT  ;Timeslice    = 1000
COMMENT  ;MinCPUPCT    = N/A
COMMENT  ;MaxCPUPCT    = N/A
```

```
************************
COMMENT
COMMENT  Workgroup     = BS_Default
COMMENT  ;Memb_Queue   = BS
COMMENT  ;Base         = 100
COMMENT  ;Limit        = 150
COMMENT  ;MinQuant     = N/A
COMMENT  ;MaxQuant     = N/A
COMMENT  ;Boost        = N/A
COMMENT  ;Timeslice    = 1000
COMMENT  ;MinCPUPCT    = N/A
COMMENT  ;MaxCPUPCT    = N/A
COMMENT
************************
COMMENT
COMMENT  Workgroup     = CS_Default
COMMENT  ;Memb_Queue   = CS
COMMENT  ;Base         = 152
COMMENT  ;Limit        = 154
COMMENT  ;MinQuant     = 1
COMMENT  ;MaxQuant     = 2000
COMMENT  ;Boost        = DECAY
COMMENT  ;Timeslice    = 200
COMMENT  ;MinCPUPCT    = N/A
COMMENT  ;MaxCPUPCT    = N/A
COMMENT
************************
COMMENT
COMMENT  Workgroup     = DS_Default
COMMENT  ;Memb_Queue   = DS
COMMENT  ;Base         = 202
COMMENT  ;Limit        = 238
COMMENT  ;MinQuant     = 2000
COMMENT  ;MaxQuant     = 2000
COMMENT  ;Boost        = DECAY
COMMENT  ;Timeslice    = 200
COMMENT  ;MinCPUPCT    = N/A
COMMENT  ;MaxCPUPCT    = N/A
COMMENT
************************
COMMENT
COMMENT  Workgroup     = ES_Default
COMMENT  ;Memb_Queue   = ES
COMMENT  ;Base         = 240
COMMENT  ;Limit        = 253
COMMENT  ;MinQuant     = 2000
COMMENT  ;MaxQuant     = 2000
COMMENT  ;Boost        = DECAY
COMMENT  ;Timeslice    = 200
COMMENT  ;MinCPUPCT    = N/A
COMMENT  ;MaxCPUPCT    = N/A
```

Use the text editor you prefer to edit the configuration file that you produced from the SHOWWG command. Use the existing workgroup specifications as a guideline to add new workgroups to the file or make any necessary changes to the existing workgroups.

While you are editing the configuration file, remember the following points:

- You cannot change or delete any of the system-defined workgroups via the NEWWG command, which is why they are preceded with the COMMENT keyword.
- You cannot create a new workgroup that has the same name as one of the system-defined workgroups, or the name NATURAL_WG, (irrespective of case; the names themselves are reserved).
- When you enter the information for a new workgroup, you must specify a workgroup name, at least one membership criterion, and the *base* and *limit* scheduling parameters.
- You may enter an entire workgroup specification in one physical record (rather than entering each parameter on a new line, as occurs in the WGFILE format. It is not allowed, however, to have two workgroup specifications in the same physical record.
- You may "comment out" a specification by inserting the COMMENT keyword before each line of the workgroup description.
- The order that the workgroups appear in the list is significant. The Workload Manager uses a "first fit" algorithm when placing processes in workgroups. The first workgroup in the list for which a process qualifies will be its destination.

To keep the modified configuration separate from the original file, save it with a unique file name. You may have occasion to restore the previous configuration, and it will be useful to have this information on hand, in the ASCII format required by the NEWWG command.

To validate the indirect file

When you use the VALIDATE option of the NEWWG command, the Workload Manager will check the indirect file for errors in syntax or semantics *without* changing the current configuration. This is a good way to pre-test the file, particularly if you have just created or edited it, since it allows you the opportunity to fix any problems before you actually need to reconfigure the workgroups on your system.

To validate the indirect file *without* changing the configuration, enter:

```
NEWWG ^filename;VALIDATE
```

Enter the name of your indirect file in place of *filename*. Also, be sure to precede the name with the caret ("^"), otherwise the Workload Manager will interpret the indirect file name as the name of a workgroup.

The Workload Manager will check the indirect file for any errors and report them to you.

To replace the configuration now

To replace the existing configuration with a new one now, enter the NEWWG command and specify the indirect file that contains the new workgroup information on the command line. In this case, the syntax of the NEWWG command is:

```
NEWWG ^filename
```

Make certain that you precede the indirect file name with the caret ("^"), otherwise the CI will interpret the indirect file as the name of a workgroup.

For example, to replace the existing configuration with the workgroups contained in the file weekend, enter:

```
NEWWG ^weekend
```

This action is *atomic*, i.e., either all workgroups are created or, if there were syntax or semantic errors, none are. A **system-wide scan** is done after all workgroups are created to determine workgroup membership for all processes on the system.

Note

To add a new workgroup to the existing configuration, you use a different syntax and enter all of the pertinent information on the NEWWG command line. For information, read "To create a new workgroup", earlier in this chapter.

To replace the configuration at a specified time

Once you have created and validated an indirect file, you can schedule the introduction of the new workgroup at a specific time. To do so, you enter the NEWWG command in a job file, and then **stream** the job. Follow these steps:

1. Using the text editor of your choice, create a job file using the sample below as a model:

   ```
   !JOB wgchange,manager.sys
   !NEWWG ^weekend
   !EOJ
   ```

2. Keep the job file with a name that is meaningful to you. It can be the same as the name of the job you created, for example wgchange.

3. Use the STREAM command to schedule the job for introduction at a specified time. For example, if it is Friday morning and you want the new configuration to take effect at six p.m. in the evening, enter:

   ```
   :STREAM wgchange; AT=18:00
   ```

To replace the configuration based on conditional criteria

You could also create a job that conditionally introduces a new workgroup configuration. For example, suppose you want to introduce a workgroup configuration that allocates a minimal amount of CPU to batch jobs, giving preference to your online data entry staff. However, suppose also that there was problem with the batch runs the night before, and they have not yet completed. In this case, you can create a job file that includes `IF` statements to check for a specified condition and, if it has been met, invokes the `NEWWG` command.

Altering a Workgroup

To make changes in the scheduling characteristics of a workgroup, you use the `ALTWG` command. Do not use the `ALTWG` command to change the membership criteria or the workgroup name. To do that requires the `NEWWG` command.

Here is the syntax of the `ALTWG` command:

```
ALTWG [WORKGROUP=]workgrp
[[;BASE=]base]  [[;LIMIT=]limit]
[[;MINQUANT=]min]  [[;MAXQUANT=]max]
[[;BOOST=]{DECAY    }]
          {OSCILLATE}
[[;TIMESLICE=]tslice]
[[;MINCPUPCT=]minpercent]  [[;MAXCPUPCT=]maxpercent]
```

For example, to set the `Program_Development` workgroup's timeslice to 500 milliseconds, enter:

```
ALTWG Program_Development ;timeslice=500
```

Or, for example, to set new base and limit priorities for the `Payroll_online` workgroup, and give the workgroup a minimum CPU percentage of 30, enter:

```
ALTWG Payroll_online ;base=190 ;limit=215; mincpupct=30
```

To set the same scheduling characteristics using positional parameters, enter:

```
ALTWG Payroll_online,190,215,,,,,30
```

Note

If you get an error message when attempting to set the minimum or maximum CPU percentages, issue a SHOWWG command to find out what other minimum and maximums have been set. For *minpercent*, the total for all workgroups cannot exceed 99 percent and, as a result, the Workload Manager will not allow you to set a minimum CPU percentage that raises this sum beyond 99. For each workgroup, the maximum CPU percentage (*maxpercent*) must be equal to or greater than *minpercent*.

To create an artificial workgroup member

An artificial member of a workgroup is a process that has been explicitly placed in the workgroup via the ALTPROC command. In general, it is preferable to let processes naturally migrate to the appropriate workgroup based on matching membership criteria. However, you may need to place a process in a workgroup when:

- You want to move a process that is starving others to a low-priority workgroup.
- You need to move a critical job that must be finished quickly to a higher-priority workgroup.
- You get a call from a user whose system performance is sluggish, and he or she urgently needs to finish something.

Here is the syntax of the ALTPROC command:

```
ALTPROC [ [PIN=] { pinspec                          }
                 { (pinspec [,pinspec ] ... )       }
        [;JOB=] { jobspec                           }
                { (jobspec [,jobspec] ... )         } ]

{ [;PRI=]pri                                        }
{ [;WG= ] { workgroup    }                          }
          { NATURAL_WG   }                          }

[ { ;TREE   } ]
  { ;NOTREE }

[ { ;USER    } ]
  { ;ANYUSER }

[;SYSTEM]
```

For example, to move the user process identified as PIN number 47 to the CS_Default workgroup, enter:

```
ALTPROC 47;WG=CS_Default
```

You can see which processes are artificial members of a workgroup with either the PROCS format of the SHOWWG command, or with the DETAIL format of the SHOWPROC command. In both cases, artifical workgroup members have a percent sign (%) next to their process identification number. For example, to display information about PIN 47, you enter:

```
SHOWPROC 47;FORMAT=DETAIL
```

Note

Avoid using the ;PRI= option of the ALTPROC command unless you have no user-defined workgroups on your system and you want to move a process to one of the other system-defined workgroups. Instead, to place a process at a specific priority, create a workgroup with the *base* and *limit* set to the values you want and then use the ALTPROC *pin*;WG= command to move the process to that workgroup.

To return a process to its natural workgroup

You also use the ALTWG command to return a process to its natural workgroup. To do so, identify the PIN of the process that you want to move, and then enter:

```
ALTPROC pin;WG=NATURAL_WG
```

Purging a Workgroup

To delete a user-defined workgroup from the current configuration, you use the PURGEWG command. You can delete one workgroup, several workgroups, or all of the user-defined workgroups on the system. You cannot delete any of the system-defined workgroups.

Here is the syntax of the PURGEWG command:

```
PURGEWG [WORKGROUP=]{ workgrpspec }
{ (workgrpspec [ ,workgrpspec ] ... ) }

[ [;ONERROR=] { CONTINUE } ]
              { QUIT     }

[ { ;CONFIRM    } ]
[ { ;NOCONFIRM  } ]
[ { ;CONFIRMALL } ]

[ { ;NOSHOW } ]
[ { ;SHOW   } ]

[ { ;SHOWERRORS   } ]
[ { ;NOSHOWERRORS } ]

[ { ;SHOWERROR   } ]
[ { ;NOSHOWERROR } ]

[ { ;PURGESCAN   } ]
[ { ;NOPURGESCAN } ]
```

The options to the PURGEWG command allow you to determine what to do in case there is an error, how much control you have over the purge operation, and how much information you see.

To delete one workgroup

To delete one user-defined workgroup on the system, enter:

```
PURGEWG WORKGROUP=workgrpspec
```

For example, to delete a workgroup named Payroll_batch, enter:

```
PURGEWG WORKGROUP=Payroll_batch
```

Or you can enter:

```
PURGEWG Payroll_batch
```

To delete all user-defined workgroups

To delete all user-defined workgroups on the system, you can issue the PURGEWG command and specify the @ wildcard character. Even though this literally translates to "all workgroups", not just user-defined workgroups, the Workload Manager will not purge any of the system-defined workgroups.

For example, suppose you have 3 user-defined workgroups on your system and, of course, the 5 system-defined workgroups. When you issue the PURGEWG @ command, you'll see the following messages:

```
:PURGEWG @

8 workgroups matched
Continue PURGEWG ? (YES/NO)y
8 selected. 3 succeeded. 5 failed.
Operation failed on some files. (CIWARN 490)
```

To defer scanning a workgroup

When you purge a workgroup, the Workload Manager immediately scans all of its member processes and moves them to other workgroups. This action is called a **purgescan**. You can choose to defer the purgescan until a later time with the NOPURGESCAN option of the PURGEWG command. Deferring a purgescan is a good idea when you are deleting multiple workgroups. Otherwise, the Workload Manager may place processes from the workgroup you just purged into another workgroup that you plan to purge subsequently.

For example, to purge the workgroup `Payroll_batch` and defer scanning, enter:

```
PURGEWG Payroll_batch;NOPURGESCAN
```

Once you have issued the `PURGEWG` command with the `NOPURGESCAN` option, the workgroup becomes "purge-pending". **Purge-pending workgroups** are those which still have member processes, and they remain in a purge- pending state until the member processes are moved via a purgescan, until you explicitly reassign them to another workgroup, or until they die.

To request a purgescan of all purge-pending workgroups

To explicitly request that Workload Manager rescan all processes belonging to purge pending workgroups, enter:

```
PURGEWG;PURGESCAN
```

4

Tuning System Performance

This chapter teaches you how to use workgroups to optimize system performance. In particular, it discusses how you can use the Workload Manager to handle predictable changes in CPU demand, how to respond to changes in the workload, and how to provide a more consistent level of service to your users.

The topics in this chapter include:

- Scheduling anticipated workload changes
- Giving one workgroup improved CPU access
- Giving one workgroup degraded CPU access
- Creating workgroups for reactive processing
- Handling a starving process
- Handling a CPU-bound process that impacts other users
- Managing the workload to provide more consistent response times
- Balancing workload during a system consolidation

Scheduling Anticipated Workload Changes

Often, system managers can predict the behavior of their system workload. You know the performance requirements at 9 a.m. differ from those at 6 p.m. and that the weekday workload isn't the same as the weekend workload. The Workload Manager allows you to plan for such changes in the workload and adjust the workgroup configuration accordingly.

To make "proactive" changes to your workgroup configuration, you can create individual configuration files for each type of workload and introduce the new configuration at any time with the `NEWWG` command. The following steps summarize this procedure. For more detailed information, refer to "Creating a New Workgroup Configuration" in Chapter 3.

1. Create a workgroup configuration file in ASCII format.

   It isn't necessary to create the file from scratch. For convenience, use the `WGFILE` format of the `SHOWWG` command to write the current configuration to a file, and then edit that file.

2. Use the `VALIDATE` option of the `NEWWG` command to check the file for errors in syntax or semantics *without* replacing the configuration.

3. When you want to change the configuration, issue the `NEWWG` command, specifying the name of the configuration file on the command line.

4. To change the configuration at a specified time, include the `NEWWG` command (that names the configuration file) in a job, and schedule the job for streaming at the appropriate time.

Responding to Unanticipated Changes in the Workload

Creating a suitable workgroup configuration that you can adjust to meet anticipated changes in usage is the most desirable way to tune system performance. However, situations may arise that require you to manage your system reactively instead of proactively. Using the Workload Manager, you can resolve system problems by adjusting the behavior of a single workgroup or a single process. The following sections present typical scenarios in which reactive changes are necessary.

To improve a workgroup's CPU access

A situation may arise in which you need to give an entire workgroup improved access to the CPU. For example, suppose it is a particularly heavy day for orders and your telephone sales representatives need improved response time.

To improve a workgroup's access to the CPU, you can use the `ALTWG` command to do any of the following:

- Increase the base and limits of the workgroup to higher priorities. Since the MPE/iX Dispatcher is priority-driven, this will give the members of this workgroup preference over lower-priority processes.
- Increase the minimum CPU percentage of this workgroup (and decrease the minimums of other workgroups). Note that this is *only* effective if the workgroup is using all of the CPU its minimum provides.
- Increase the maximum CPU percentage of the workgroup (and decrease the maximums of other workgroups). Note that this is *only* effective if the workgroup has already reached its original maximum.

To degrade a workgroup's CPU access

A situation can arise in which you need to decrease the priority of an entire workgroup. For example, suppose a batch run did not complete the night before and is now impacting the response time of interactive users. Or, perhaps a group of users is receiving 0.25 second response time when a 0.5 second response would be sufficient, making additional CPU available to other processes.

To degrade a workgroup's access to the CPU, you can use the `ALTWG` command to do any of the following:

- Decrease the base and limits of the workgroup to lower priorities. Since the MPE/iX Dispatcher is priority-driven, this will give other workgroups with a higher range of priorities preference over this one.
- Decrease the minimum CPU percentage of this workgroup. Note that this is *only* effective if the workgroup is already using the CPU its minimum provides. Lowering the CPU minimum from 25% to 20% will have no effect if the workgroup is only consuming 15%.
- Impose a maximum CPU percentage on the workgroup, or lower the existing maximum CPU percentage, such that the workgroup is constrained to run within a smaller percentage of the CPU allocation than it can consume.

To modify CPU access for a single process

There may be occasion when you need to single out a process and move it to a higher or lower priority. For example:

- You may need to increase a process' priority if an online user has a critical task and needs faster response time, or if a batch job needs to finish quickly.
- You may need to decrease a process' priority if it is in an infinite loop that is critical and cannot be killed, or if a CPU intensive process is performing a task that isn't time-critical.

To handle one process that is causing problems, do the following:

1. Use the `NEWWG` command to create a high-priority workgroup that you will use to temporarily hold processes that require more of the CPU. For example, to create a `Hipri_procs` workgroup with the base and limit priorities set to 152, enter:

   ```
   :NEWWG Hipri_procs; memb_logon="special,manager.sys";&
   base=152;limit=152; position=AS_Default
   ```

   In this example, the membership criteria requires that someone log on as `manager.sys`, using the job or session name `special`. Since the user name and account is password-protected, it is highly unlikely that anyone other than you could logon and gain access to this workgroup by naturally meeting the membership criterion. (Instead of using your own user identity in the membership criterion, you could also create a logon that does not exist on your system.)

   This command includes the `POSITION` parameter to place the workgroup at the end of the list of user-defined workgroups, which minimizes scanning.

2. Use the `NEWWG` command to create a low-priority workgroup that you will use to temporarily hold processes that are using too much of the CPU. For example, to create a `Lowpri_procs` workgroup with the base and limit priorities set to 255, enter:

   ```
   :NEWWG Lowpri_procs; memb_logon="special,manager.sys";&
   base=255;limit=255; position=AS_Default
   ```

   As in step one, this membership criteria is sufficiently restrictive to prevent others from becoming natural members of this workgroup.

3. If necessary, enter the `SHOWPROC` command to identify the process whose CPU access you need to modify.

4. Use the `ALTWG` command to move the process to the appropriate workgroup. For example, to give a process improved CPU access, move it to the `Hipri_procs` workgroup:

   `ALTPROC` *pin*`;WG=Hipri_procs`

Or, for example to move a CPU-bound process that is impacting other users to a low priority, enter:

```
ALTPROC pin;WG=Lowpri_procs
```

Note

Another way to guarantee improved access to the CPU is to define a larger minimum CPU value, as well as high base and limit priorities, for the `Hipri_procs` workgroup. Then, move processes to that workgroup on an "as needed" basis.

Providing More Consistent Response Times

The Workload Manager features can be used to ensure more consistent response times for users. For example, you might need to meet a specific Service Level Agreement (SLA) with users, minimize performance complaints, or facilitate capacity planning. The grouping of processes into workgroups gives the system manager the ability to partition the system into groups with similar needs. Altering the scheduling characteristics of those workgroups provides the control over CPU access, which in turn helps determine response time.

Caution

It is *critical* to understand that CPU access is just one component of response time. The Workload Manager can help handle this aspect, but cannot handle problems with disk access speeds, memory constraints, network latency and availability, or the other components of response time.

In controlling CPU access, you can either control the workgroup needing the consistent response times, or identify the other workgroup(s) whose behavior leads to inconsistent response times and control those workgroups.

To control a single workgroup

In handling a single workgroup, you can change the base and limit priorities, change the rate of priority decay (by adjusting the quantum), or assign CPU minimum and/or maximum percentages. Which control is most effective depends on the characteristics of the processes you want to control. It may be the case that placing the workgroup at priorities 160-170 gives consistent 1-second response time to those users. Alternatively, you may need to set a minimum CPU percentage of 20% to the workgroup in order to ensure 1-second response time.

To control other workgroups

If processes in one workgroup (`workgroup_1`) are experiencing inconsistent response times, it may be due to the influences of another workgroup (`workgroup_2`). For example, `workgroup_2` may be set to a higher priority and contain processes that perform long transactions, leading to increased response times for the processes in `workgroup_1`. To handle the situation, you can:

- Move the processes in `workgroup_2` that perform long transactions to a lower-priority workgroup.
- Move the entire workgroup (`workgroup_2`) to a lower priority.
- set a CPU maximum for `workgroup_2`, restricting the amount of CPU it can consume.
- If it is a single process that is disrupting the response times of others, move it to a lower-priority workgroup.

Balancing Workload During System Consolidation

The Workload Manager can facilitate the process of consolidating multiple source systems onto a target system. You can use workgroups to help you plan and execute the consolidation, and to manage the final consolidated system.

To partition the workload on the target system

One typical concern regarding consolidation is the limited amount of control the system manager has over the target system. Before the introduction of the Workload Manager, five scheduling subqueues were available on *each* of the source systems, and *only* five scheduling subqueues were available on the target system. As a result, the scheduling subqueues on the consolidated system contain a larger numer and varieity of processes than did the source systems, which decreased the system manager's ability to control CPU scheduling.

With the introduction of the Workload Manager, you can now define multiple workgroups to represent the users of each of the source systems. For example, suppose you are consolidating three systems. In this case, you could create workgroups to represent the CS, DS, and ES processes from each one, for a total of nine workgroups on the target system. This preserves the partitioning that had been available with the physical separation of the source systems.

Alternatively, you can use the Workload Manager to define workgroups that more naturally reflect the needs of the combined user population. Perhaps data entry clerks had been in the CS subqueue of several source systems and now you can combine them into a single workgroup on the target system. You might collect similar batch jobs into a common workgroup. Or, you could separate users who were once forced to share the CS subqueue into distinct workgroups. The scheduling characteristics of the workgroups on the target system can be adjusted to result in the CPU access that the system manager requires to achieve desired performance.

To manage user expectations

Another area of concern relates to the effect of the consolidation itself on user's experience and expectations of system performance.

Consider a situation where you are consolidating Systems A, B, and C onto System D over a period of time. You plan to bring over System A on the first weekend, System B on the second weekend, and System C on the final weekend. During the first week on the target system, while the first set of users is running alone on System D, the response time and throughput are excellent. When they are joined by the users from System B, their performance may degrade. Once all three systems are combined on the target, the System A users may actually complain about their performance. Even though it may be better than what they had on System A, the performance degraded as additional users were added to System D.

How can the Workload Manager be used to help this situation? The first set of users on the target system grew dissatisfied because they had become accustomed to the better performance when they had exclusive use of the target system. To solve this problem, you can restrict the amount of CPU available to users. For example, constrain System A users to 30% of the target system so that they will experience, from the onset, the performance that will result when the entire consolidation is complete.

This example is obviously simplified. You may not wish to divide the target system up evenly among the users from the three source systems. Perhaps one set of users is more important and requires more of the CPU. Alternatively, the consolidation may raise the larger concern of how to ensure satisfactory coexistence of competing workloads from the various source systems once they have been consolidated. The Workload Manager gives you the tools you need to create an effective workgroup configuration, monitor the performance, and make adjustments as necessary to ease the consolidation process.

5

Commands Reference

This chapter describes the four new commands you use to create, modify, display and delete workgroups: `NEWWG`, `ALTWG`, `SHOWWG`, and `PURGEWG`. It also documents the four existing commands affected by the introduction of workgroups on the HP 3000, `ALTPROC`, `SHOWPROC`, `SHOWQ` and `TUNE`, and notes the new features and changes needed to accommodate them.

The eight commands are listed in alphabetical sequence.

ALTPROC

Changes characteristics of the specified processes. Currently, a process' priority, queue attribute, and workgroup may be changed. (Native Mode)

SYNTAX

```
ALTPROC [ [ PIN= ] { pinspec                          } ]
        [          { (pinspec [ ,pinspec ] ... )      } ]
        [ ;JOB=  ] { jobspec                          }
        [          { (jobspec [ ,jobspec ] ... )      }

[ [ ;PRI= ] pri                                       ]
[ [ ;WG=  ] { workgrp                                 }
            { NATURAL_WG                              }

[ { ;TREE   } ]
[ { ;NOTREE } ]

[ { ;USER   } ]
[ { ;ANYUSER} ]

[ ;SYSTEM ]
```

PARAMETERS

pinspec

The process(es) you want to alter. This is a required parameter, unless you specify *jobspec*. If you omit both, you will get an error.

The *pinspec*, expressed [#p]pin, is a Process Identification Number (PIN). If *pinspec* is 0, then the caller's pin is used. To alter system processes, you must have SM capability and specify the `SYSTEM` option.

`NOTREE` is the default for all *pinspec* target processes, and can be overridden with the `TREE` option.

The `USER` and `ANYUSER` options do not apply to *pinspec*.

jobspec

The name of the job or session whose processes are to be altered. A *jobspec* can be any of the following, jobnumber, username, @S, @J, or @.

- The jobnumber must be in the form of either *#Jnnn* or *#Snnn*.
- The username must be in the form *user*[.*account*]. If there is more than one job/session matching the same username, they will all be altered.
- Wildcards have the following meanings:

  @S - all sessions

  @J - all jobs

  @ - all sessions and jobs

The USER and ANYUSER options apply only to *jobspec* and only if *jobspec* is wildcarded. The USER option, which is the default, alters only processes matching the user's name, while the ANYUSER option alters all processes matching the wildcarded *jobspec*. For example, if the user's name is STEVE.UI and you enter the command shown below, then only job processes logged on as STEVE.UI are altered.

```
:ALTPROC job=@j;pri=cs
```

However, if you add anyuser to the same command as shown below, then all job processes are altered.

```
:ALTPROC job=@j;pri=cs;anyuser
```

TREE is the default for all *jobspec* target processes, and can be overridden with the NOTREE option.

The SYSTEM option is ignored for all *jobspec* target processes.

The *jobspec* is optional as long as a *pinspec* is supplied. If both are omitted, an error is reported.

pri — The queue for the process. If omitted, the priority is unchanged.

Caution

Exercise extreme caution when altering a process's priority, scheduling queue attribute, or workgroup membership. Such a change can significantly impact system performance.

Note

Avoid using the ;PRI= option to explicitly change a process. If you have created user-defined workgroups that have ;MEMB_QUEUE as membership criteria, use of the ;PRI= option may change the workgroup. Instead, use either the ;WG=*workgrp* or ;WG=NATURAL_WG option, explained below, to move target processes into existing workgroups.

Using ;WG= to explicitly change a workgroup should be a temporary measure, and used rarely. Instead, adjust workgroup membership criteria to ensure that desired processes become natural members of the workgroup.

If you specify both the ;WG= and ;PRI= in the ALTPROC command line, you will get an error.

The *pri* value may be one of the following:

- A scheduling queue value {BS,CS,DS,ES} sets the queue attribute of the target process(es). If a user-defined workgroup does not capture the process, then the process will fall into to the corresponding system-defined default workgroup at the base priority (subject to decay as it consumes CPU). To assign a scheduling queue value, you must have OP capability.
- A queue manager value {BM,CM,DM,EM} sets the queue attribute of the target process(es). If a user-defined workgroup does not capture the process, then the process will fall into the corresponding system-defined default workgroup at the base priority (non-decayable). To assign a queue manager value, you must have SM capability.
- An absolute priority {*nnn*} sets the priority of the process to the specified value that will not decay. The workgroup of the process will not be changed (the process will have the same timeslice value). Note that the priority specified need not fall between the base and limit priorities of the workgroup. To assign an absolute priority value, you must have SM capability.

If you do not have SM capability, then your `MAXPRI` value represents the highest priority that you can assign a process. A warning appears when the specified priority exceeds `MAXPRI`. `MAXPRI` is ignored for System Manager (SM) capability.

workgrp A workgroup value moves the target process(es) to the specified workgroup. A process moved in this manner is considered an **artificial member** of the workgroup (the process was placed in workgroup explicitly, rather than naturally by meeting the membership criteria specified for the workgroup).

A process remains an artificial member of its assigned workgroup until either the workgroup is purged or its explicit assignment is changed (via ALTPROC or an AIF call). An artificial member is not affected by a system-wide scan or by the changing of its process attributes used to determine workgroup membership. A workgroup specification requires SM capability and can only be used to modify the workgroup assignment of user processes.

You cannot specify both the ;WG= and ;PRI= in the ALTPROC command line. Workload Manager users should use ;WG= instead of ;PRI=.

NATURAL_WG — The natural workgroup specification {NATURAL_WG} releases one or more process(es) from their explicit workgroup assignment, allowing them to migrate to their natural workgroup. A natural workgroup specification requires SM capability.

TREE — This option alters each process specified as well as all of its descendants. TREE is the default for all *jobspec* target processes. If you specify both TREE and SYSTEM, you will see a warning that TREE will be ignored.

NOTREE — This option alters only the processes specified. Descendant processes will not be altered. NOTREE is the default for all *pinspec* target processes.

USER — The USER option applies only when *jobspec* is wildcarded. It alters only processes matching the user's name. USER is the default.

ANYUSER — The ANYUSER option applies only when *jobspec* is wildcarded. It alters all *jobspec* target processes, regardless of their owners.

SYSTEM — Use the SYSTEM option if the target process specified in *pinspec* is a system process. SM capability is required for the SYSTEM option. SYSTEM is ignored for all *jobspec* processes and when you specify a workgroup or natural workgroup. If you specify both SYSTEM and TREE, you see a warning that TREE will be ignored.

Caution

Exercise extreme care when altering system processes since doing so can significantly degrade system efficiency.

OPERATION

To execute the ALTPROC command, you must have System Supervisor (OP) or System Manager (SM) capability. SM capability is necessary to alter system processes, for the WG= option, for certain specifications to the PRI option, and to increase a process' priority above MAXPRI.

You may issue the ALTPROC command from a session, job, program, or while in BREAK. Pressing (Break) aborts the execution of this command.

EXAMPLE

To alter process 605, and its current descendants, so that their priorities execute within the DS_Default workgroup, enter:

```
:ALTPROC #p605; tree; wg=DS_Default
```

To alter process 605, and its current descendants, so that their scheduling queue attribute is DS, enter:

```
:ALTPROC #p605; tree; pri=DS
```

The outcome of this command is not necessarily identical to the outcome achieved with the previous command. If the system was configured with a user-defined workgroup that captured the processes (MEMB_QUEUE=DS and a match on other specified membership attributes as well), then the processes would be a member of the user-defined workgroup rather than the DS_Default workgroup.

To alter all job processes to the CS_Default workgroup, enter:

```
:ALTPROC job=@j; wg=CS_Default; anyuser
```

To return the processes modified by the previous example to their natural workgroup(s), enter:

```
:ALTPROC job=@j; wg=NATURAL_WG; anyuser
```

To alter all job processes matching the user's name to the CS_Default workgroup, enter:

```
:ALTPROC job=@j; wg=CS_Default; user
```

To alter the current process' priority so that it behaves like a CS queue manager (SM capability required), enter:

```
:ALTPROC 0;pri=CM
```

To alter all processes logged on as mgr.payroll to linear 155 (SM capability required), enter:

```
:ALTPROC job=mgr.payroll; pri=155
```

To alter the queue attribute of pins 150, 247, 211 to be ES, enter:

```
:ALTPROC (150,#p247,211); pri=ES
```

Related Information

Commands	SHOWPROC, TUNE, SHOWQ, NEWWG, ALTWG, PURGEWG, SHOWWG
Manuals	*MPE/iX Intrinsics Reference Manual* (32650-90028)

ALTWG

Alters the scheduling characteristics of an existing workgroup. (Native Mode)

SYNTAX

```
ALTWG [WORKGROUP=]workgrp
[[;BASE=]base] [[;LIMIT=]limit]
[[;MINQUANT=]min] [[;MAXQUANT=]max]
[[;BOOST=]{DECAY }] {OSCILLATE}
[[;TIMESLICE=]tslice]
[[;MINCPUPCT=]minpercent] [[;MAXCPUPCT=]maxpercent]
```

Note

To alter the membership criteria of a workgroup, you must use the NEWWG command.

PARAMETERS

Note

ALTWG follows the CI convention that positional parameters are allowed until the first high-level keyword is specified.

workgrp — The name of the workgroup whose scheduling characteristics you are modifying. This is a required parameter. *Use of wildcards is not allowed.*

base — An integer from 150 to 255 specifying the highest priority at which processes executing in the workgroup begin their Dispatcher transactions. Priority is inversely related to the integer: a higher-priority process has a lower number. While the full range is provided for compatability, avoid setting the base priority between 150 and 152, since user processes running at a higher priority than 152 can adversely affect system performance.

limit — An integer specifying the lowest priority which processes executing in the workgroup can attain. Priority is inversely related to the integer: a higher-priority process has a lower number. The *limit*, which can range from 150 to 255, must be greater than or equal to the *base*.

min — The minimum number of milliseconds that a process may use the CPU before its priority is reduced. The *min* is a lower bound for the quantum value, which determines the rate of priority decay for processes within the workgroup. Values range between 1 and 32,767, inclusive.

max — The maximum number of milliseconds that a process may use the CPU before its priority is reduced. The *max* is an upper bound for the quantum value, which determines the rate of priority decay for processes within the workgroup. The value of *max* must be greater than or equal to the value of *min*. Values range between 1 and 32,767, inclusive.

`DECAY` — Sets the workgroup to the default decay behavior associated with circular scheduling subqueues. If set, a process decays normally to the *limit* priority and returns to the *base* priority when the Dispatcher transaction is complete. `DECAY` is the default boost property.

`OSCILLATE` — Sets the workgroup to oscillate behavior. If set, a process returns to the *base* priority once its priority has decayed to the *limit* of the workgroup, even if it has not completed a Dispatcher transaction.

tslice — The timeslice is the maximum number of milliseconds a process in the workgroup can hold the CPU before returning to the Scheduler to have its priority recalculated. Values must be multiples of 100, with a minimum value of 100 and a maximum value of 32700.

minpercent — The minimum percentage of time that the CPU is available to the workgroup's member processes. Within the target workgroup, this CPU time is allocated according to the processes' priorities. The workgroup is guaranteed this minimum percentage even if higher-priority processes in other workgroups are ready to run.

If the processes in the workgroup do not require as much CPU time as their guaranteed minimum, that time will be available to processes in other workgroups. The default value for this optional workgroup characteristic is that the workgroup is not guaranteed any minimum CPU allocation and that CPU allocation is based on process priority only.

maxpercent The maximum percentage of time that the CPU is available to the workgroup's member processes. The workgroup will be limited to this maximum percentage of time *even if no other process in another workgroup is ready to run*. In other words, a workgroup will be limited to this maximum percentage even if the system must remain idle for the remainder of the time.

The default value for this optional workgroup characteristic is that there is no maximum which would prevent a workgroup's processes from getting the CPU time their priorities warrant.

OPERATION

Each workgroup consists of three components, a name, membership criteria, and scheduling characteristics. The membership criteria (`MEMB_LOGON`, `MEMB_PROGRAM` and `MEMB_QUEUE` parameters), determine process assignment, which occurs on six occasions:

- at process creation
- whenever you change one of the process attributes on which membership can be based
- when you explicitly move a process to a workgroup via the `ALTPROC` command
- when you purge a workgroup, and the subsequent purgescan redistributes processes to remaining workgroups
- when you add a new workgroup to the current workgroup configuration
- when the current workgroup configuration is replaced

Membership conflicts are resolved by assigning a process to the **first** workgroup for which it qualifies. (Workgroups are maintained in an ordered list in which position is important. For more information, read the discussion of the `POSITION` parameter of the `NEWWG` command, later in this chapter.) Since the Workload Manager uses a first-fit algorithm, other workgroups are not considered once a match is found. For this reason, it is important to arrange the workgroups in your configuration from those with the most precise membership criteria to those with more general membership criteria.

The scheduling characteristics define the scheduling policies which govern processes within the workgroup. The MPE/iX Dispatcher is priority-driven, allocating a CPU to the highest priority process that is ready to run. The MPE/iX Scheduler determines process priorities in accordance with workgroup scheduling characteristics. Each workgroup has a *base* and a *limit*, defining the priority range for processes within that workgroup. Processes will begin their Dispatcher transactions at the *base* priority and decay towards the *limit* priority as they consume system resources. The process priority will never drop below the *limit* of the workgroup. If the boost property of the workgroup is set to oscillate, the process priority will be reset to the *base* priority when it decays to the *limit* priority. Processes completing Dispatcher transactions (typically via a terminal read) are reset to the *base* priority.

Each of the five system-defined workgroups represents one of the five scheduling subqueues, AS, BS, CS, DS and ES. Their **only** membership criteria is the *scheduling queue attribute*. For example, the AS_Default workgroup has as its membership criteria `MEMB_QUEUE=AS`. These workgroups appear last in the ordered list so that the Scheduler can guarantee that all processes will be assigned to a workgroup.

The `ALTWG` command allows a user to alter only the scheduling characteristics of an existing workgroup, not the membership criteria. As a result, there is no need for a system-wide scan since workgroup membership does not change. (To change the name or membership criteria of an existing user-defined workgroup, you must use `NEWWG`.)

`ALTWG` is similar in function to `TUNE`. In fact, if you need to adjust the scheduling characteristics of the CS_Default, DS_Default, and ES_Default workgroups, you may use the `TUNE` command (instead of `ALTWG`) to do so. Altering the scheduling characteristics of a workgroup will be deferred while a scan is in progress.

You may issue the `ALTWG` command from a session, job, program or in BREAK. Pressing Break has no effect on this command. `ALTWG` requires System Supervisor (OP) or System Manager (SM) capability.

Note

Altering the scheduling characteristics of a workgroup will affect each of the member processes. That is, the Scheduler might need to adjust each process' priority attribute. The time required to make these priority adjustments is related to the number of processes in the workgroup.

EXAMPLE

To set the Program_Development workgroup's *base* to 158, *limit* to 168, and boost priority to decay, enter:

```
:ALTWG Program_Development, 158, 168, , , decay
```

To set the Program_Development workgroup's timeslice to 500 milliseconds, enter:

```
:ALTWG Program_Development ;timeslice=500
```

Related Information

Commands: NEWWG, PURGEWG, SHOWWG, TUNE, SHOWQ, ALTPROC, SHOWPROC

Manuals: *MPE/iX Intrinsics Reference Manual* (32650-90028)

NEWWG

Creates a new, user-defined workgroup either directly, via command line input, or indirectly, through a file. (Native Mode)

SYNTAX

```
NEWWG ^filename [;VALIDATE]
```

Or:

```
NEWWG [WORKGROUP=]workgrp

{ [;MEMB_LOGON=]logon
  [;MEMB_PROGRAM=]program_file
  [;MEMB_QUEUE=]queue_attribute }

[;BASE=]base  [;LIMIT=]limit

[[;MINQUANT=]min]  [[;MAXQUANT=]max]

[[;BOOST={DECAY|OSCILLATE}]]

[[;TIMESLICE=]tslice]

[[;MINCPUPCT=]minpercent]  [[;MAXCPUPCT=]maxpercent]

[[;POSITION=]existingwg]
```

Note

Misuse of this command can significantly degrade system operating efficiency.

PARAMETERS

Use the first parameter (*^filename*) and the first option (VALIDATE) to specify an indirect file with the NEWWG command.

^filename

The name of the indirect file which contains workgroup specifications. This file can be used to replace the current workgroup configuration with that specified in the indirect file. The file name must be preceded with the "^" (caret) character.

When you specify an indirect file, the current user-defined workgroup configuration is *replaced* with the valid workgroup specification in the file. This action is *atomic*, i.e., either all workgroups are created or, if there were syntax or semantic errors, none are. A system-wide scan is done after all workgroups are created to determine workgroup membership for all processes on the system.

Note that this is an indirect file, *not* a command file. A command file would issue a NEWWG command for every new workgroup created and force a system-wide scan after each. Furthermore, creation would not be atomic.

Refer to the File Format section below for a description of the parameters valid for use within the indirect file (they include all of the parameters valid for the command-line specification except `POSITION`, since `POSITION` is actually the workgroup's position within the indirect file).

`;VALIDATE` Use this option to check the indirect file for the correct syntax and report any errors. The workgroup configuration in the file is **not** invoked. This option lets you ensure that when you subsequently issue the NEWWG command specifying this indirect file, the operation will not fail due to errors in syntax or semantics. Do not include this option in the indirect file or you will get an error.

Use the following parameters to create a new workgroup and add it to the current configuration by entering information on the command line.

workgrp The name of the workgroup you are creating. This is a required parameter.

The workgroup name follows the convention for CI variables and Job Control Words (JCWs) and can be a maximum of 255 alphanumeric characters or underscores, where the first character cannot be numeric. The user-specified name (including case) is preserved, though comparisons are case-insensitive. All 255 characters are significant.

The following names, expressed in any case (all upper, all lower, or mixed) are unavailable: AS_Default, BS_Default, CS_Default, DS_Default, ES_Default, and Natural_wg.

The following parameters for logon, program file and queue attribute consistute the membership criteria for the workgroup. You must specify at least one of these parameters. Multiple specifications are permitted within a parameter (with commas as delimiters). Paremeters that you do not specify are assumed matches.

`OR` is used between values within each parameter and, `AND` is used across specified parameters. That is, a process must match one of the specifications for each of the specified parameters. Membership conflicts are resolved by assigning the process to the first matching workgroup defined. That is, a first-fit algorithm is used and, once a match is found, no further workgroups are considered.

logon Specifies the job/session, user, and account name of potential workgroup members. The job/session name is optional, but if specified, you must enclose the entire logon string in double quotes (" "). The user and account names are required. You may use wildcards in any part of the logon string. The format is shown below:

```
MEMB_LOGON = (logonname [,logonname]...)
```

Where *logonname* is:

```
"job/session name, username.acctname"
```

or

```
username.acctname
```

program_file Specifies the program files of potential workgroup members. The *program_file* must be a fully qualified MPE/iX file name or absolute Hierarchical File System (HFS) file name. You may use wildcards in MPE/iX filenames. The format is shown below:

```
MEMB_PROGRAM = (program_file [,program_file]..)
```

Note that for HFS file names, comparison is case-sensitive. Thus, if the file exists in the MPE name space, you must specify it in upper case, for example, `/SYS/PUB/WMTEST` instead of `/sys/pub/wmtest`.

queue_attribute Specifies the traditional queue attribute. The `MEMB_QUEUE` parameter is provided for compatibility. It enables the system manager to collect processes into workgroups without having to change their existing logon priority or job card. Any current specification or alteration of scheduling queue attribute (via `HELLO`, `JOB`, `LINK`, `ALTPROC`, `AIFPROCPUT`, `GETPRIORITY`) will continue to set the queue attribute, but this attribute can be used to determine the workgroup membership of the process. As new job streams or tasks are created, you may wish to avoid specifying scheduling queue and allow workgroup membership to be determined by the other categories such as `MEMB_LOGON` and `MEMB_PROGRAM`.

You may enter the following values for *queue_attribute*: AS, BS, CS, DS, and ES.

```
MEMB_QUEUE = (queue_attribute [,queue_attribute]..)
```

base — An integer from 150 to 255 specifying the highest priority at which processes executing in the workgroup begin their Dispatcher transactions. Priority is inversely related to the integer: a higher-priority process has a lower number. While the full range is provided for compatability, avoid setting the base priority between 150 and 152, since user processes running at priorities greater than 152 can adversely affect system performance. The *base* is a required scheduling characteristic.

limit — An integer specifying the lowest priority which processes executing in the workgroup can attain. Priority is inversely related to the integer: a higher-priority process has a lower number. The *limit*, which can range from 150 to 255, must be greater than or equal to the *base*. It is a required scheduling characteristic.

min — The minimum number of milliseconds that a process may use the CPU before its priority is reduced. The *min* is a lower bound for the quantum value, which determines the rate of priority decay for processes within the workgroup. Values range between 1 and 32767. The minimum quantum is an optional scheduling characteristics with a default value of 1 millisecond.

max — The maximum number of milliseconds that a process may use the CPU before its priority is reduced. The *max* is an upper bound for the quantum value, which determines the rate of priority decay for processes within the workgroup. The value of *max* must be greater than or equal to the value of *min*. Again, values range between 1 and 32767. The maximum quantum is an optional scheduling characteristics with a default value of 2000 milliseconds.

`DECAY` — Sets the workgroup to the default decay behavior associated with circular scheduling subqueues. If set, a process decays normally to the *limit* priority and returns to the *base* priority when the Dispatcher transaction is complete. `DECAY` is the default boost property.

`OSCILLATE` — Sets the workgroup to oscillate behavior. If set, a process returns to the *base* priority once its priority has decayed to the *limit* of the workgroup, even if it has not completed a Dispatcher transaction.

tslice The timeslice is the maximum number of milliseconds a process in the workgroup can hold a CPU before returning to the Scheduler to have its priority recalculated. Values must be multiples of 100, with a minimum value of 100 and a maximum value of 32700. The timeslice is an optional scheduling characteristic with a default value of 200 milliseconds.

minpercent The minimum percentage of time that the CPU is available to the workgroup's member processes. Within the target workgroup, this CPU time is allocated according to the processes' priorities. The workgroup is guaranteed this minimum percentage even if higher-priority processes in other workgroups are ready to run.

If the processes in the workgroup do not require as much CPU time as their guaranteed minimum, that time will be available to processes in other workgroups. The default value for this optional workgroup characteristic is that the workgroup is not guaranteed any minimum CPU percent and that CPU allocation is based on process priority only.

maxpercent The maximum percentage of time that the CPU is available to the workgroup's member processes. The workgroup will be limited to this maximum percentage of time *even if no other process in another workgroup is ready to run*. In other words, a workgroup will be limited to this maximum percentage even if the system must remain idle for the remainder of the time.

The default value for this optional workgroup characteristic is that there is no maximum which would prevent a workgroup's processes from getting the CPU time their priorities warrant.

existingwg Workgroups are maintained in an ordered file. Therefore, the `POSITION` parameter, *existingwg*, establishes the position of the new workgroup within the set of existing user-defined workgroups. The *existingwg* value is the name of any existing, user-defined workgroup or the `AS_Default` system-defined workgroup. The new workgroup is inserted before the existing workgroup. Specifying `AS_Default` positions the new workgroup at the end of the list of user-defined workgroups, immediately preceding the system-defined workgroups.

The `POSITION` specification is optional. If omitted, the new workgroup is appended to the end of the list of existing user-defined workgroups.

Note

You may wish to define workgroups with "dummy" membership criteria that would never match process attributes. Such workgroups will not have natural members, but you can use them as the target of an `ALTPROC;WG=` command when you need to quickly move a process that is affecting system performance. Such workgroups should be placed *after* all user-defined workgroups to minimize the number of workgroups that must be scanned to determine process workgroup membership.

FILE FORMAT

The `NEWWG` command accepts an indirect file, containing the specifications for creating user-defined workgroups, as input. The indirect file should be an ASCII file that is temporary or permanent and have fixed or variable length records. When you specify an indirect file, its workgroup configuration *replaces* the existing workgroup configuration on your system.

Workgroup creation begins after all specifications within the file have passed a syntax and semantic check. Furthermore, the system will consider the creation an atomic operation, i.e, either all workgroups within the file are created or none are. Once all workgroups are created, a system-wide scan is performed to determine workgroup membership.

The specification for an individual workgroup is given below. The parameters shown match those described when using the command line. The `POSITION` parameter is not valid within the indirect file, since a workgroup's position in the ordered list of workgroups is determined by its position within the file.

```
Workgroup      = workgrp
;Memb_Logon    = logon
;Memb_Program  = program_file
;Memb_Queue    = queue_attribute
;Base          = base
;Limit         = limit
;MinQuant      = min
;MaxQuant      = max
;Boost         = {DECAY | OSCILLATE}
;Timeslice     = tslice
;Mincpupct     = minpercent
;Maxcpupct     = maxpercent
```

You must specify the workgroup name, at least one membership parameter, and the *base* and *limit* parameters. Multiple specifications are permitted within a parameter. Use commas as delimiters and an "&" or (Return) to indicate the continuation of a specification onto a new line. Parameters that you do not specify are assumed matches. For example, if you only specified `MEMB_QUEUE=(CS)`, the workgroup would capture all processes with the CS queue attribute (with any logon, running any program). Since only the `MEMB_QUEUE` category has been specified, the other categories are assumed matches.

Only `BASE` and `LIMIT` are required scheduling parameters. The others are optional and will be set to default values unless explicitly changed.

The example above shows each parameter on a new line. However, the entire workgroup specification may reside in one physical record. The only restriction is that you cannot have two workgroup specifications in the same physical record.

You may "comment out" a specification by using the `COMMENT` keyword, as shown below. Characters appearing on the same line and after the `COMMENT` keyword are ignored.

```
COMMENT  Workgroup     = Old_Finance_WG
COMMENT  ;Memb_Logon   = @.TEST
COMMENT  ;Memb_Program = EDITOR.PUB.SYS
COMMENT  ;Memb_Queue   = ES
COMMENT  ;Base         = 200
COMMENT  ;Limit        = 230
COMMENT  ;MinQuant     = 200
COMMENT  ;MaxQuant     = 1000
COMMENT  ;Boost        = DECAY
COMMENT  ;Timeslice    = 400
COMMENT  ;Mincpupct    = 20
COMMENT  ;Maxcpupct    = 30
```

Note

Before you invoke the `NEWWG` command and specify an indirect file to replace workgroups, you may want to use the `SHOWWG` command and redirect the output to create an ASCII file that describes the existing configuration. For example, to create an ASCII file named "currwg" enter:

```
:SHOWWG @; format=wgfile > currwg
:SAVE currwg
```

Substitute a file name of your own choosing, and make certain that it is unique. If you want to use a name that's longer than 8 characters, for example, `current_workgroups`, precede it with `./` to indicate HFS syntax.

OPERATION

The system manager creates workgroups to reflect a partitioning of the system workload and control the allocation of the CPU to user processes on the system. There is essentially no hard limit to the maximum number of workgroups that a system may have. The system will always be configured with five default workgroups, which exist to support system processes and to provide backward compatibility.

Each workgroup consists of three components, a name, membership criteria, and scheduling characteristics. The membership criteria (`MEMB_LOGON`, `MEMB_PROGRAM` and `MEMB_QUEUE` parameters), determine process assignment. Workgroup assignment is made at each process create, and whenever one of the process attributes on which membership can be based is changed, and when workgroups are purged or new workgroups created. Membership conflicts are resolved by keeping the workgroups in an ordered list. A process is assigned to the first workgroup whose membership criteria is matched. That is, a first-fit algorithm is used. Once a match is found, no other workgroups are considered. Therefore, order your workgroup specifications from those with the most precise membership criteria to those with more general membership criteria.

The scheduling characteristics define the scheduling policies which govern processes within the workgroup. The MPE/iX Dispatcher is priority-driven, giving a CPU to the highest priority process that is ready to run. Each workgroup has a *base* and a *limit*, defining the priority range for processes within that workgroup. Processes will begin their Dispatcher transactions at the *base* priority and decay towards the *limit* priority as they consume system resources. The process priority will never drop below the *limit* of the workgroup. If the boost property of the workgroup is set to oscillate, the process priority will be reset to the *base* priority when it decays to the *limit* priority. Processes completing Dispatcher transactions (typically via a terminal read) are reset to the *base* priority.

Each of the five system-defined workgroups represents one of the five scheduling subqueues, AS, BS, CS, DS and ES. Their **only** membership criteria is the *scheduling queue attribute*. For example, the AS_Default workgroup has as its membership criteria `MEMB_QUEUE=AS`. These workgroups appear last in the ordered list so that the Scheduler can guarantee that all processes will be assigned to a workgroup.

The user-defined workgroups can control all user processes. System processes, however, are placed in one of the five default workgroups and handled appropriately by the Scheduler. The system manager can customize the characteristics of the default workgroups to reflect the CPU scheduling needs of the various components of the system workload.

The SHOWWG command offers a format option, WGFILE, whose output has a format suitable as input to NEWWG. You may use CI I/O redirection to place the output from SHOWWG into a file. For example, SHOWWG @;FORMAT=WGFILE > *filename*, and then issue the SAVE command to save it.

Creating a new workgroup can affect the workgroup assignment of existing processes. The NEWWG command forces the Scheduler to scan all processes on the system and adjust their workgroup membership as necessary. This is referred to as a system-wide scan. As a result, there is a performance advantage in using an indirect file, since it allows you to define multiple workgroups and, once all workgroups have been created, performs a single system-wide scan.

The NEWWG command may be issued from a session, job, program or in BREAK. Pressing (Break) has no effect on this command. NEWWG requires System Supervisor (OP) or System Manager (SM) capability.

The following default settings are established when the system is booted from the system disk (a START RECOVERY or START NORECOVERY), unless the user has customized a workgroup configuration.

Default Workgroup Settings After a System Start

Setting:	AS_Default	BS_Default	CS_Default	DS_Default	ES_Default
base	13	100	152	202	240
limit	99	150	200	238	253
min	N/A	N/A	1	2000	2000
max	N/A	N/A	2000	2000	2000
boost	N/A	N/A	DECAY	DECAY	DECAY
tslice	1000	1000	200	200	200
mincpupct	N/A	N/A	N/A	N/A	N/A
maxcpupct	N/A	N/A	N/A	N/A	N/A

Note

Processes within the workgroup are not constrained to have their priorities fall within the specified base and limit values. N/A refers to the fact that the scheduling characteristic is Not Applicable for the system-defined workgroup. For example, processes within the AS_Default or BS_Default workgroup (usually system processes) do not experience decay. Therefore, these workgroups do not support the concept of a quantum or boost priority. Also, CPU percentages are not available for the default workgroups; they apply only to user-defined workgroups.

EXAMPLE

The following example creates a user-defined workgroup named Program_Development:

```
:NEWWG Program_Development; memb_program= (editor.pub.sys,&
qedit.@.@, hpedit.@.@);memb_logon=("nm@,@.mytest"); base=&
160; limit= 170; boost= oscillate
```

Or, using positional parameters instead of keywords, you would enter:

```
:NEWWG Program_Development, "@nm@,@.mytest",&
(editor.pub.sys,qedit.@.@,hpedit.@.@),,160,170,,,oscillate
```

The membership criteria **must** appear in the order MEMB_LOGON, MEMB_PROGRAM and MEMB_QUEUE when the keywords are not specified. In the preceding example, since MEMB_QUEUE isn't specified, a comma is used as a placeholder. The ampersand character (&) appears at the end of the command lines in the examples above to indicate continuation. If you type the command on one single line (so that it wraps), omit the ampersand.

This workgroup has a base priority of 160, a limit of 170, the default minimum and maximum quantum of 1 and 2000 milliseconds, an oscillating boost property, and the default timeslice of 200 milliseconds. The membership criteria requires a program of editor.pub.sys or qedit.@.@ or hpedit.@.@, AND a logon equal to nm@,@.mytest. This workgroup will be appended to the end of the user-defined workgroups.

To create a second workgroup, ahead of the workgroup in the previous example, enter:

```
:NEWWG Program_Test; memb_logon= (@.test); base= 175;&
limit= 190; boost= oscillate; position= Program_Development
```

This workgroup has a base priority of 175, a limit of 190, the default minimum and maximum quantum of 1 and 2000 milliseconds, an oscillating boost property, and the default timeslice of 200 milliseconds. The membership criteria requires a logon equal to @.test.

To validate the indirect file spec1.workgrp.system for correctness (without invoking the changes), enter:

```
:NEWWG ^spec1.workgrp.system; validate
```

To have NEWWG replace the current workgroup configuration with the workgroups specified in spec1.workgrp.system, enter:

```
:NEWWG ^spec1.workgrp.system
```

Related Information

Commands	ALTWG, PURGEWG, SHOWWG, TUNE, SHOWQ, ALTPROC, SHOWPROC
Manuals	*MPE/iX Intrinsics Reference Manual* (32650-90028)

PURGEWG

Purges the specified user-defined workgroup(s). (Native Mode)

SYNTAX

```
PURGEWG [WORKGROUP=]{ workgrp }
       { (workgrp [,workgrp] ... ) }

[ [;ONERROR=] { CONTINUE } ]
              { QUIT     }

[ { ;CONFIRM    } ]
  { ;NOCONFIRM  }
  { ;CONFIRMALL }

[ { ;NOSHOW } ]
  { ;SHOW   }

[ { ;SHOWERRORS   } ]
  { ;NOSHOWERRORS }

[ { ;PURGESCAN   } ]
  { ;NOPURGESCAN }
```

PARAMETERS

workgrp
: The user-defined workgroup(s) that you want to purge. This is a required parameter. You may use wildcards, but if you specify @ (to indicate all workgroups), only the user-defined workgroups are purged since you cannot purge the five system-defined workgroups.

CONTINUE
: Allows PURGEWG to continue until the end of the list is reached, regardless of errors. CONTINUE is the default option.

QUIT
: Quits the execution of PURGEWG when it encounters an error and sets the CIERROR variable to the last execution error.

CONFIRM
: Verifies the *workgrp* parameter by requiring you to validate the purge during command execution. Valid responses are "YES" or "NO". If you respond "YES", the PURGEWG command is executed. Pressing Break at the prompt is equivalent to responding "NO". CONFIRM is the default for sessions, unless the *workgrp* designates a single workgroup.

NOCONFIRM
: Continues the purge without verification from the user. NOCONFIRM is the default for jobs or if the *workgrp* designates a single workgroup.

`CONFIRMALL` When you are purging multiple workgroups from a session, use the `CONFIRMALL` option to verify each workgroup before the purge is executed. You may respond with any of the following:

"Y" or "YES"	To purge the workgroup
"N", "NO", or Return	To retain the workgroup
"Q", "QUIT", or Break	To stop the `PURGEWG` command

The `CONFIRMALL` option is ignored in jobs and when you are purging a single workgroup.

`NOSHOW` Suppresses the display of each successfully purged workgroup. `NOSHOW` is the default.

`SHOW` Displays the name of each successfully purged workgroup.

`SHOWERRORS` Displays each lower-level error which prevents a workgroup from being deleted. The name of the workgroup is shown, followed by the error message. By default lower-level errors are not displayed. You may also enter this option in the singular form, i.e. `SHOWERROR`.

`NOSHOWERRORS` Suppresses the display of low-level errors. `NOSHOWERRORS` is the default. You may also enter this option in the singular form, i.e. `NOSHOWERROR`.

`PURGESCAN` Instructs the Workload Manager to rescan processes belonging to purge pending workgroups *after* executing the `PURGEWG` command. This is the default.

`NOPURGESCAN` Instructs the Workload Manager to defer the rescan of processes belonging to purge pending workgroups until explicitly requested to do so. To explicitly request a rescan, issue the command `PURGEWG ;PURGESCAN`.

OPERATION

Use the `PURGEWG` command to purge existing user-defined workgroups. The command requires a *workgrp* which can be one workgroup or a list of workgroups. In addition, you may use wildcard characters in *workgrp* to specify multiple workgroups. If you specify "@", only the user-defined workgroups are purged since the five system-defined workgroups cannot be purged.

The PURGEWG command handles user prompting through three options, CONFIRM, CONFIRMALL, and NOCONFIRM. The CONFIRM option requests verification of the PURGEWG command as a whole. That is, CONFIRM only asks for verification of the *workgrp* parameter. The CONFIRMALL option provides user prompting for each workgroup. The NOCONFIRM option overrides all verification. Since the job environment does not support user prompting, the CONFIRM and CONFIRMALL options are ignored.

By default the PURGEWG command does not display the name of each purged workgroup. To override the default, use the SHOW option.

The ONERROR keyword lets you specify the PURGEWG command's response if it encounters an error while trying to purge an individual workgroup in a pattern match situation. (This can occur, for example, if you try to purge one of the default workgroups which is not allowed). Use the CONTINUE option to continue the execution of the PURGEWG command until the end of the list is reached. Use QUIT, to quit command execution at the point where PURGEWG encountered the error.

When purging a list of workgroups, one of three results is possible:

- The purge succeeded on all workgroups, and as a result, the CIERROR variable is unchanged.
- The purge failed on some of the workgroups. As a result, the CIERROR variable is set to the value CIWARN 490.
- The purge failed on all of the workgroups. As a result, the CIERROR variable is set to the value CIERR 491.

When you specify ONERROR=QUIT, CIERROR is set to the last error which stopped the execution. For example, if there are no user-defined workgroups that end in `default` on your system, and you enter the command `purgewg @default` in an attempt to purge the system-defined workgroups, the CIERROR is set to 12205. The text of this error message is `Cannot delete a system-defined workgroup`.

In addition, the following variables are set only when you specify a list of workgroups:

- HPNUMSELECTED: Contains the number of workgroups selected, which matches the number in the *workgrp* unless you responded "NO" when prompted to verify the purge of one or more workgroups.
- HPNUMSUCCEEDED: Contains the number of workgroups successfully purged.
- HPNUMFAILED: Contains the number of workgroups that did not get purged.

When PURGEWG is discontinued before it reaches the end of the list, the difference between HPNUMSELECTED and HPNUMSUCCEEDED + HPNUMFAILED indicates the number of workgroups that were selected but not purged.

When a workgroup is purged, the Workload Manager needs to rescan the affected member processes. The cost of such a rescan depends upon the number of processes and workgroups involved. There are situations in which you may want to defer the rescan. In terms of the workgroup, the system cannot complete the purge until all member processes have found a new workgroup. A workgroup in such a state is considered to have a purge pending. The scan of processes assigned to purge-pending workgroups is a subset of a system-wide scan. That is, a system-wide scan checks every process on the system. A purge-pending scan only checks processes that are assigned to purge-pending workgroups. While a purge-pending scan is performed by default, the NOPURGESCAN option allows you to defer the rescan.

The CI supports a second syntax, which you use to explicitly initiate a purge-pending scan without requiring the purging of a workgroup:

```
PURGEWG [;PURGESCAN]
```

Warning

Misuse of the ability to defer a scan of processes assigned to purge-pending workgroups could significantly impact system performance because processes remain members of purge-pending workgroups.

Logically, a workgroup in the purge-pending state no longer exists. That is, the workgroup cannot accept new members. However, the workgroup physically remains until either its last member has died or has been moved to another workgroup, or until a scan is performed. Since you may want to create a new workgroup that uses the name of a purge-pending workgroup, the system automatically renames such workgroups when they enter the purge-pending state. The new name becomes the previous name, prepended with a "~". The SHOWWG command displays the workgroup with its new name so that users know that the workgroup is in the purge-pending state.

The purging of a user-defined workgroup is deferred while a system-wide or purge-pending scan is in progress.

The PURGEWG command may be issued from a session, job, program or in BREAK. Pressing Break aborts execution of this command. PURGEWG requires System Supervisor (OP) or System Manager (SM) capability.

EXAMPLE To purge all user-defined workgroups, you can enter the commands as shown in either of the two samples below:

```
:PURGEWG @; noconfirm

2 workgroups matched
2 workgroups selected.  2 workgroups succeeded.  0 workgroups failed.
:

:PURGEWG @; noconfirm; show

2 workgroups matched
 User_WG1
 User_WG2
2 workgroups selected.  2 workgroups succeeded.  0 workgroups failed.
:
```

To purge just the `User_WG1` workgroup, enter:

```
:PURGEWG User_WG1
```

The Workload Manager does not display any confirmation messages.

To purge the multiple workgroups by specifying a list, enter each workgroup name on the command line, separated by commas, and enclosed in parenthesis. For example:

```
:PURGEWG (User_WG1, User_WG2)
2 workgroups matched
CONTINUE PURGE ? (YES/NO) yes
2 workgroups selected.  2 workgroups succeeded.  0 workgroups failed.
:
```

To purge multiple workgroups by specifying a wildcard, enter the partial workgroup name and the appropriate wildcard character. For example:

```
:PURGEWG User_WG#
2 workgroups matched
CONTINUE PURGE ? (YES/NO) yes
2 workgroups selected.  2 workgroups succeeded.  0 workgroups failed.
:
```

To purge the User_WG1 and User_WG2 workgroups and request confirmation, enter:

```
:PURGEWG User_WG#; confirmall
2 workgroups matched
 User_WG1 ? (N/Y) Y
 User_WG2 ? (N/Y) Y
2 workgroups selected.  2 workgroups succeeded.  0 workgroups failed.
:
```

Related Information

Commands: NEWWG, ALTWG, SHOWWG, TUNE, SHOWQ, ALTPROC, SHOWPROC

Manuals: *MPE/iX Intrinsics Reference Manual* (32650-90028)

SHOWPROC

Displays information about the specified process(es). (Native Mode)

SYNTAX

```
SHOWPROC [ [PIN=] { pinspec                                  } ]
         [        { (pinspec [ ,pinspec ] ... )              } ]

         [ [;JOB=] { jobspec                                 } ]
         [         { (jobspec [ ,jobspec ] ... )             } ]

         [ [;FORMAT=] { SUMMARY } ]
         [            { DETAIL  } ]

         [ { ;TREE   } ]
         [ { ;NOTREE } ]

         [ { ;USER    } ]
         [ { ;ANYUSER } ]

         [ { ;SYSTEM } ]

         [ { ;TRUNC   } ]
         [ { ;NOTRUNC } ]
```

PARAMETERS

pinspec

The process that you want to see.

The *pinspec*, expressed [#p]pin, is a Process Identification Number (PIN). Specifying *pinspec* is optional and has no default; see *jobspec*.

An ordinary user may show processes matching their own user and account names (those which "belong to" the user) by specifying 0 as the *pinspec*. A user with SM or OP capabilities may show any process on the system. A user with SM capability (the system manager) may see system processes by specifying the `SYSTEM` option.

`NOTREE` is the default for all *pinspec* target processes, and can be overridden with the `TREE` option.

The `USER` and `ANYUSER` options do not apply to *pinspec*.

jobspec

The name of the job or session whose processes you want to display. A *jobspec* can be any of the following: jobnumber, username, @S, @J, or @. A *jobspec* is optional and defaults to the user's current job ID, for example, `#!HPJOBTYPE!HPJOBNUM`.

The jobnumber must be in the form *#Jnnn* or *#Snnn*. SM or OP capability is required to specify another user's job or session number. The username must be in the form *user*[.*account*]. SM or OP capability is required to specify another user's username. If there is more than one job or session under the same username, all are displayed.

You can use the wildcard symbol `@` in the following ways:

`@S`	Shows process information for all sessions
`@J`	Shows process information for all jobs
`@`	Shows process information for all sessions and jobs

An ordinary user can only see their own processes, even when *jobspec* is wildcarded. For example, if the user name is JEFF.MFG and you enter the command as shown below, then only processes for jobs logged on as JEFF.MFG are displayed.

```
:SHOWPROC job=@J
```

On the other hand, if the user STEVE.UI (who has OP or SM capability) enters the command shown below, then all processes for all jobs on the system are displayed.

```
:SHOWPROC job=@J
```

If the user STEVE.UI only wants to see his own job processes, he must enter:

```
:SHOWPROC job=@J; user
```

The `USER` option, and its counterpart option, `ANYUSER`, are described below.

The `SYSTEM` option is ignored for all *jobspec* target processes.

`TREE` is the default for all *jobspec* target processes, and can be overridden with the `NOTREE` option.

`SUMMARY` This format displays a subset of a process' attributes. These include the subqueue name, process priority, CPU time, execution state, associated `JOB` or `SESSION` number, `PIN` (indented to show tree structure), program name, and `INFO=string`, if any (or command step if the process is CI.PUB.SYS). `SUMMARY` is the default format.

DETAIL	This format displays a more comprehensive set of the attributes associated with a process.
TREE	This option displays each process specified, as well as all of its descendents. TREE is the default for all *jobspec* target processes.
NOTREE	This option displays only the process specified. No information appears for the process's descendants. NOTREE is the default for all *pinspec* target processes.
SYSTEM	The SYSTEM option is required if the target process from *pinspec* is a system process. It displays system processes as well as descendant user processes. SM capability is required. SYSTEM is ignored for all *jobspec* processes.
USER	The USER option filters output when *jobspec* is wildcarded by displaying only processes matching the user's name. USER is the default for users without OP and SM capability.
ANYUSER	This option defeats the filtering of the wildcarded *jobspec* and displays all matching processes. SM or OP capability is necessary to specify ANYUSER, and users with these capabilities get ANYUSER by default. OP or SM users may reduce the SHOWPROC output to just their own processes by using the USER option.
TRUNC	The TRUNC option truncates output records that would exceed the record width of $STDLIST for the user. A $ replaces the last character of the line to signify truncation. TRUNC is the default option.
NOTRUNC	This option displays output records in their full form. As a result, output from the command may wrap around the display.

OPERATION

The SHOWPROC command displays information about processes except lockwords, which are never displayed. By default, the processes shown are the root CI and its descendents (TREE option). Any user may issue this command. Users with OP or SM capability may see information for processes belonging to other users. SM users may also see system processes via the SYSTEM option.

Any user may issue the SHOWPROC command and see information about all processes that belong to them. A process "belongs" to a user if one or more of the following conditions exists:

1. the process is within the user's logon job/session
2. the process' user and account names match the user's user and account names *and* the system's JOBSECURITY is set to LOW
3. the user has OP or SM capability.

If rule 1 or 2 applies or the user has SM capability then all information (except lockwords) is visible. Otherwise, only the Command Interperter (CI) command and/or program names are shown. That is, the parameters of a CI command and the INFO= string passed to a program are not visible.

When SHOWPROC is executed in a job, regardless of capabilities and process ownership, only the CI command name and program are displayed.

If you specify both the ;PIN= and ;JOB= parameters, information for the list of pins will precede the information for the list of jobs. Duplicate specifications are not detected.

SHOWPROC may be issued from a Session, Job, Program, or in BREAK. Pressing (Break) aborts the execution of this command.

The fields displayed are described below. The field's width, in characters, is shown within parentheses. A "v" indicates that the field has a variable size width.

CPUTIME (8): CPUTIME is consumed in hh:mm:ss or m:ss.mls. A pair of asterisks (**) appears in the hours field when hours overflows. The three-character "mls" sub-field holds milliseconds.

JOBNUM (6): The job or session number for the process.

LOGON (v): The job/session, user, and account name associated with this process.

PARENT (5): Process Identification Number for the process' parent (decimal). This field is unique to the DETAIL format. The DETAIL format displays PARENT so that process relationships can be determined. A zero indicates that the process does not have a parent (for example, PROGEN).

PIN (5): Process Identification Number for the process (decimal). The SUMMARY format indents the PIN column by two spaces for each child process so that you can clearly see a process' descendants. The DETAIL format precedes the pin with a percent sign (%) to indicate that the process is an artificial member of its workgroup, and does not indent the display.

PRI (5) The priority at which the process is currently executing. A lower numeric value indicates a higher priority. It also indicates whether the process is linear, runs with fixed priority (L), or is decayable (D). This field is unique to the DETAIL format.

PROGRAM (v):	The file name of the program the process is executing.
QUEUE (v):	The scheduling queue attribute associated with this process. The QUEUE field is unique to the DETAIL format.
QPRI (5):	A combination of SUBQUEUE and PRIORITY which appears as Qnnn[*]. Q is a single character abbreviation of the process' scheduling queue attribute. The nnn is the process' priority, and * indicates that this process is a system process. The QPRI field is unique to the SUMMARY format.
STATE (5):	The execution state of the process, which can be one of the following: ■ BLKIO blocked for terminal write or control. ■ WAIT generic process block, usually waiting for a message. ■ BLKCB blocked for control block. ■ BLKMM blocked for memory manager. ■ READY ready to execute (or executing).
STEP (v):	The command that the displayed CI process is currently executing. This field is not shown for non-CI processes.
WORKGROUP (v):	The workgroup of which the process is a member. WORKGROUP appears as [%]name, where % indicates that the process is an artificial member of the workgroup, and name is the workgroup name. A process becomes an artificial member when it is explicitly placed into the workgroup via ALTPROC or AIFPROCPUT instead of naturally meeting the membership criteria of the workgroup.

On the next page is a sample output of the DETAIL format. In this example, pin 2 is a system mode process, running linearly at priority 142. Pin 99 is a user mode process running linearly at priority 160. Pin 121 is a user mode process that is an artificial member of the "Payroll_Online" workgroup

```
:SHOWPROC pin=(2,99,121,188);format=detail;system

 PIN  PARENT    PRI    CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----   --------  -----   ------  --------------
 2  1          142 L   7:23.687  WAIT            (LOAD.PUB.SYS)

LOGON         :
PROGRAM       :  LOAD.PUB.SYS
QUEUE         :  BS
WORKGROUP     :  BS_Default

**********************


 PIN  PARENT   PRI    CPUTIME   STATE   JOBNUM   (PROGRAM) STEP
-----  ------  -----   --------  -----   ------  --------------
 99  68        160 L   0:05.020  BLKIO    S45    (QEDIT.PUB.SYS)

LOGON         :  NMTEST,SLC.MYTEST
PROGRAM       :  QEDIT.PUB.SYS
QUEUE         :  BS
WORKGROUP     :  Program_Development

**********************


 PIN  PARENT    PRI    CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----   --------  -----   ------  --------------
 121 97       158 D   0:12.045  READY    J51     :tdp "text report"

LOGON         :  JREPORT,GREG.MYTEST
PROGRAM       :  TDP.PUB.SYS
QUEUE         :  DS
WORKGROUP     :  %Payroll_Online


**********************

 PIN  PARENT    PRI    CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----   --------  -----   ------  --------------
 188 101      100 D   0:04.200  WAIT     S56     (TDP.PUB.SYS) text test1

LOGON         :  CMTEST,DOUG.MYTEST
PROGRAM       :  TDP.PUB.SYS
QUEUE         :  BS
WORKGROUP     :  BS_Default
```

Below is a sample output of the default SUMMARY format.

```
:SHOWPROC #P54; tree; trunc

QPRI  CPU       STATE JOBNUM  PIN   (PROGRAM) STEP

C152  0:12.999  WAIT  S12     54    :tdp "text myfile"
C152  0:02.000  WAIT  S12       38    (TDP.PUB.SYS) text myfile
C152  0:01.030  READY S12         67    (FCOPY.PUB.SYS)from=foo.pub.sys;to=b$


:SHOWPROC #P54; tree; notrunc

QPRI  CPU       STATE JOBNUM  PIN   (PROGRAM) STEP

C152  0:12.999  WAIT  S12     54    :tdp "text myfile"
C152  0:02.000  WAIT  S12       38    (TDP.PUB.SYS) text myfile
C152  0:01.030  READY S12         67    (FCOPY.PUB.SYS)from=foo.pub.sys;to=ba
r;new
```

EXAMPLE To display a summary of information for all non-system processes in the current job/session, enter:

```
:SHOWPROC
```

To display a summary of information for PIN 42, enter:

```
:SHOWPROC #p42
```

To display a summary of information for PIN 42 and all of its descendants, enter:

```
:SHOWPROC #p42; tree
```

To display the detail information for PIN 42, enter:

```
:SHOWPROC #p42; format= detail
```

To display a summary of information for all processes (requires SM capability), enter:

```
:SHOWPROC 1 ;system ;tree
```

To display a summary of information for all non-system processes that are jobs (requires SM or OP capability), enter:

```
:SHOWPROC job=@j; anyuser
```

To display a summary of information for PINs 150, 247, and 211, enter:

```
:SHOWPROC (150,#p247,211)
```

To display a summary of information for all non-system processes logged on as MGR.PAYROLL (requires SM or OP capability), enter:

```
:SHOWPROC job=mgr.payroll
```

To display a summary of information for all non-system processes belonging to Job 2 or logged on as ME.AP (requires SM or OP capability), enter:

```
:SHOWPROC job=(#j2,me.ap)
```

To display the detail information for all non-system processes in the current job/session, enter:

```
:SHOWPROC detail
```

To display the detail information for all non-system processes on the system (requires SM or OP capability), enter:

```
:SHOWPROC job=@; format= detail
```

Related Information

Commands	TUNE, ALTPROC, SHOWQ, NEWWG, ALTWG, PURGEWG, SHOWWG
Manuals	*MPE/iX Intrinsics Reference Manual* (32650-90028)

SHOWQ

Displays scheduling data for all processes and the scheduling characteristics of the CS, DS and ES scheduling subqueue(s). (Native Mode)

SYNTAX

```
SHOWQ [;ACTIVE][;STATUS]
```

PARAMETERS

ACTIVE	Displays only the processes currently running or those about to run. This is the right-hand portion of the display. The STATUS lines are printed last.
STATUS	Reduces the output from SHOWQ to the final status lines of display (base and limit priorities, quantum bounds).

OPERATION

The process scheduling and subqueue information appears in two major columns: DORMANT and RUNNING. RUNNING processes are those that currently require the CPU in order to continue, or that will require it in the immediate future. CPU time is automatically allocated to the highest priority process that is ready to run. DORMANT processes are those waiting on longer-term events.

On occasion, a process appears in more than one column, indicating that it was changing state when you executed SHOWQ.

As the default, SHOWQ lists dormant and running processes and the scheduling characteristics of the CS, DS, and ES subqueues. However, the ACTIVE and STATUS options permit you to filter the SHOWQ output which, on large systems, may display hundreds of live processes.

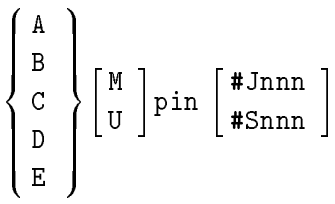
Use the ACTIVE option to display running processes and the scheduling characteristics of the CS, DS, and ES scheduling subqueues. Use the STATUS option to display just the scheduling characteristics of the CS, DS, and ES subqueues. (Note that the ACTIVE output appears when both options are specified, since status information is a subset of the active information.)

Below is an example of the two-column output produced by the SHOWQ command. The symbols that may appear in such a listing are explained in the remainder of the discussion.

```
DORMANT              RUNNING
Q PIN  JOBNUM       Q PIN  JOBNUM

A  1                C M163 #S263
B  2                C U215 #S256
B  3
A  4
D U29  #J30
C M37  #S234
C M55  #S248
```

Each entry in the three columns displays the following information for a single process; the meaning is explained below.



A	The queue attribute of the process is AS
B	The queue attribute of the process is BS
C	The queue attribute of the process is CS
D	The queue attribute of the process is DS
E	The queue attribute of the process is ES
M	This is a job or session main process
U	This is a user process
pin	Process identification number, a decimal
J *nnn*	Job number: a process executing in a batch job
S *nnn*	Session number: a process executing from a session

The process identification number (*pin*) may appear with or without an M or U label. Processes without an M or U label are system processes.

In addition, SHOWQ prints the scheduling characteristics currently in effect. In the example below, QUEUE is the scheduling subqueue and BASE, LIMIT, MIN QUANTUM, MAX QUANTUM, BOOST and TIMESLICE are scheduling values set by the TUNE command. MIN and MAX quantums are bounds for the quantums and ACTUAL quantum is the current quantum value.

```
                          ------QUANTUM------
QUEUE   BASE   LIMIT   MIN    MAX     ACTUAL   BOOST   TIMESLICE
-----   ----   -----   ---    ---     ------   -----   ---------
CQ      152    200       1    2000    200      DECAY   200
DQ      202    238    2000    2000    2000     OSC     200
EQ      240    253    2000    2000    2000     DECAY   200
```

You may issue the SHOWQ command from a session, job, program, or in BREAK. Pressing Break aborts the execution of this command. SHOWQ requires System Supervisor (OP) capability.

Note

The MPE/iX Scheduler now supports the workgroup concept. However, backward compatibility is maintained through five default workgroups created by the system. The scheduling characteristics of the CS_Default, DS_Default, and ES_Default workgroups mimic those of the CS, DS, and ES scheduling subqueues. In fact, the information displayed for the CS, DS, and ES scheduling subqueues is the same information as that for the default workgroups.

Please refer to the NEWWG and SHOWWG commands for more detail.

Since SHOWQ displays limited information regarding workgroup processes, Workload Manager users should use the SHOWWG and SHOWPROC commands rather than SHOWQ. Non-Workload Manager users may choose to use these commands if they prefer the format for viewing the default workgroups.

EXAMPLE

To display the active processes and the current scheduling subqueue characteristics, enter:

```
:SHOWQ;ACTIVE

DORMANT              RUNNING
Q PIN  JOBNUM        Q PIN  JOBNUM

                     C M163 #S263
                     C U215 #S256


                      ------QUANTUM------
QUEUE  BASE  LIMIT  MIN   MAX    ACTUAL  BOOST  TIMESLICE
-----  ----  -----  ---   ---    ------  -----  ---------
 CQ    152   200    1     2000   200     DECAY  200
 DQ    202   238    2000  2000   2000    OSC    200
 EQ    240   253    2000  2000   2000    DECAY  200
```

Related Information

Commands: TUNE, ALTPROC, SHOWPROC, NEWWG, ALTWG, PURGEWG, SHOWWG

Manuals: *MPE/iX Intrinsics Reference Manual* (32650-90028)

SHOWWG

Displays scheduling and process data pertaining to the specified workgroup(s). (Native Mode)

SYNTAX

```
SHOWWG [ [WORKGROUP=] { workgrp                                  }  ]
                      { (workgrp [ ,workgrp ] ... )              }

[                     { SUMMARY }  ]
[ [;FORMAT=]          { WGFILE  }  ]
[                     { PROCS   }  ]
[                     { DETAIL  }  ]

[ { ;TRUNC   }  ]
[ { ;NOTRUNC }  ]
```

PARAMETERS

workgrp

The name of the workgroup(s) for which information is to be displayed. Wildcards are supported. "@" matches to all workgroups, both user-defined and system-defined.

If no workgroup is specified, `@` (all workgroups) is assumed. This is the default.

Note that workgroups are displayed in the order they fall in the ordered list.

`SUMMARY`

The `SUMMARY` format displays a one-line summary of the attributes associated with the specified workgroup(s). This includes the scheduling characteristics and whether the workgroup has a purge pending. `SUMMARY` is the default format.

`WGFILE`

The `WGFILE` format displays the membership criteria and scheduling characteristics for the specified workgroups. The output generated is suitable as input to `NEWWG`. Note that while "@" matches to all workgroups, the values of the system-defined and purge-pending workgroups are preceeded by comment characters because they are not suitable input for `NEWWG`. To direct the current configuration of user-defined workgroups into an indirect file use `FORMAT=WGFILE >` *filename*.

`PROCS`

The `PROCS` format displays the member processes and certain process attributes.

`DETAIL`

The `DETAIL` format displays the attributes associated with the specified workgroup(s). This includes the membership criteria, scheduling characteristics, and member processes.

TRUNC	Truncates lines that would exceed $STDLIST for the user. A $ appears as the last character of the line to signify truncation. TRUNC is the default option for the SUMMARY, PROCS, and DETAIL formats. This option is ignored with the WGFILE format.
NOTRUNC	Allows names to appear in their full form. As a result, output from the command will wrap around the display. This option is ignored with the WGFILE format. For the SUMMARY format, NOTRUNC applies to the line length. That is, if the length of the SUMMARY line displayed exceeds the line length of the user's $STDLIST device, characters wrap to the next line.

OPERATION

Use the SHOWWG command to display attributes of the specified workgroup(s) in one of four output formats, SUMMARY, WGFILE, PROCS, and DETAIL.

The *workgrp* parameter allows for wildcarding. SHOWWG displays the workgroups in the same order that the system traverses the workgroups (for example, when performing workgroup assignment).

SHOWWG may be issued from a session, job, program, or in BREAK. Pressing Break aborts the execution of the command. SHOWWG requires System Supervisor (OP) or System Manager (SM) capability.

The SUMMARY format displays a one-line summary of the attributes associated with the specified workgroup(s). This includes the scheduling characteristics and whether the workgroup has a purge pending. Below is a sample output:

```
:SHOWWG @; FORMAT=SUMMARY

                                -----QUANTUM-----          TIME              CPU %
   WORKGROUP          BASE LIMIT   MIN   MAX  ACTUAL BOOST  SLICE   PROCS  MIN MAX
------------------- ---- -----  -----  ---- ------ -----  ------  -----  --- ---
~Program_Developm$  160  170    1000   1000 1000   DECAY  300     63      10  15
Payroll_Online      152  200    1000   1000 1000   OSC    300     433     30  40
Payroll_Batch       180  230    1000   1000 1000   DECAY  300     210     20 100
AS_Default          13   99     N/A    N/A  N/A    N/A    1000    45      N/A N/A
BS_Default          100  150    N/A    N/A  N/A    N/A    1000    73      N/A N/A
CS_Default          152  200    1      2000 576    DECAY  200     320     N/A N/A
DS_Default          202  238    2000   2000 2000   DECAY  200     217     N/A N/A
ES_Default          240  253    2000   2000 2000   DECAY  200     81      N/A N/A
```

Note

Processes within the workgroup need not have their priorities fall within the specified base and limit values. N/A refers to the fact that the scheduling characteristic is Not Applicable for the system-defined workgroup. For example, processes within the AS_Default or BS_Default workgroup (usually system processes) do not experience decay. Therefore these workgroups do not support the concept of a quantum or boost priority. Similarly, since CPU percentages do not apply to the default workgroups, you will see an N/A displayed in those columns.

The fields displayed with the `SUMMARY` format are described below. The field's width, in characters, is shown within parentheses. A "v" indicates that the field has a variable size width.

`WORKGROUP` (19): The workgroup name, which is always truncated to 18 characters. A `$` is appended to the workgroup name (as the nineteenth character) to indicate truncation.

`BASE` (3): An integer specifying the priority at which processes executing in the specified workgroup begin their Dispatcher transactions.

`LIMIT` (3): An integer specifying the lowest priority which processes executing in the specified workgroup can attain.

`MIN` (5): The minimum number of milliseconds that a process may use the CPU before its priority is reduced.

`MAX` (5): The maximum number of milliseconds that a process may use the CPU before its priority is reduced.

`ACTUAL` (5): The actual quantum value for the workgroup. The quantum is the average number of milliseconds of CPU processes within the specified workgroup required to complete their Dispatcher transactions. Processes will decay in priority when their CPU consumption exceeds the workgroup quantum.

`BOOST` (5): Indicates the priority decay behavior associated with the workgroup.

`TIMESLICE` (5): The maximum number of milliseconds a process in the workgroup can hold a CPU before returning to the Scheduler to have its priority recalculated.

`PROCS` (5): The number of member processes currently assigned to the specified workgroups.

CPU MIN (3):	The minimum CPU percentage currently established for the workgroup.
CPU MAX (3):	The maximum CPU percentage currently established for the workgroup.

The WGFILE format displays the membership criteria and scheduling characteristics for the specified workgroup(s). The output generated is suitable as input to NEWWG and is therefore never truncated. Below is an example of the WGFILE format.

```
:SHOWWG @; format=wgfile


Workgroup      = Program_Development
;Memb_Logon    = @.TEST,"NM@,@.MYTEST"
;Memb_Program  = EDITOR.PUB.SYS, QEDIT.@.@, HPEDIT.@.@
;Base          = 160
;Limit         = 170
;MinQuant      = 1000
;MaxQuant      = 1000
;Boost         = DECAY
;Timeslice     = 300
;Mincpupct     = 10
;Maxcpupct     = 15


**********************
COMMENT
COMMENT  The following workgroup is purge-pending.
COMMENT
COMMENT  WORKGROUP   = ~Test_DB
COMMENT  ;Memb_Logon = @.TEST,"NM@,@.MYTEST"
COMMENT  ;Base       = 170
COMMENT  ;Limit      = 180
COMMENT  ;MinQuant   = 1000
COMMENT  ;MaxQuant   = 1000
COMMENT  ;Boost      = OSCILLATE
COMMENT  ;Timeslice  = 200
COMMENT  ;Mincpupct  = 0
COMMENT  ;Maxcpupct  = 15


**********************
```

```
Workgroup     = Payroll_Online
;Memb_Program = PAYROLL.@.PRAPP
;Memb_Queue   = CS
;Base         = 152
;Limit        = 200
;MinQuant     = 1000
;MaxQuant     = 1000
;Boost        = OSCILLATE
;Timeslice    = 300
;Mincpupct    = 30
;Maxcpupct    = 40


*************************

Workgroup     = Payroll_Batch
;Memb_Program = PAYROLL.@.PRAPP
;Memb_Queue   = DS, ES
;Base         = 182
;Limit        = 230
;MinQuant     = 1000
;MaxQuant     = 1000
;Boost        = DECAY
;Timeslice    = 300
;Mincpupct    = 20
;Maxcpupct    = 100


*************************
COMMENT
COMMENT  The following workgroup is purge-pending.
COMMENT
COMMENT  WORKGROUP     = ~Payroll_Online
COMMENT  ;Memb_Program = PAYROLL.@.PRAPP
COMMENT  ;Memb_Queue   = CS
COMMENT  ;Base         = 152
COMMENT  ;Limit        = 200
COMMENT  ;MinQuant     = 1000
COMMENT  ;MaxAuant     = 1000
COMMENT  ;Boost        = OSCILLATE
COMMENT  ;Timeslice    = 300
COMMENT  ;Mincpupct     = 30
COMMENT  ;Maxcpupct     = 40


*************************
```

```
COMMENT  The following are system-defined (default) workgroups.
COMMENT
COMMENT
COMMENT  Workgroup   = AS_Default
COMMENT  ;Memb_Queue = AS
COMMENT  ;Base       = 13
COMMENT  ;Limit      = 99
COMMENT  ;MinQuant   = N/A
COMMENT  ;MaxQuant   = N/A
COMMENT  ;Boost      = N/A
COMMENT  ;Timeslice  = 1000
COMMENT  ;Mincpupct  = N/A
COMMENT  ;Maxcpupct  = N/A


*************************
COMMENT
COMMENT  Workgroup   = BS_Default
COMMENT  ;Memb_Queue = BS
COMMENT  ;Base       = 100
COMMENT  ;Limit      = 150
COMMENT  ;MinQuant   = N/A
COMMENT  ;MaxQuant   = N/A
COMMENT  ;Boost      = N/A
COMMENT  ;Timeslice  = 1000
COMMENT  ;Mincpupct  = N/A
COMMENT  ;Maxcpupct  = N/A


*************************
COMMENT
COMMENT  Workgroup   = CS_Default
COMMENT  ;Memb_Queue = CS
COMMENT  ;Base       = 152
COMMENT  ;Limit      = 200
COMMENT  ;MinQuant   = 200
COMMENT  ;MaxQuant   = 2000
COMMENT  ;Boost      = DECAY
COMMENT  ;Timeslice  = 200
COMMENT  ;Mincpupct  = N/A
COMMENT  ;Maxcpupct  = N/A


*************************
```

```
COMMENT
COMMENT  Workgroup   = DS_Default
COMMENT  ;Memb_Queue = DS
COMMENT  ;Base       = 202
COMMENT  ;Limit      = 238
COMMENT  ;MinQuant   = 2000
COMMENT  ;MaxQuant   = 2000
COMMENT  ;Boost      = DECAY
COMMENT  ;Timeslice  = 2000
COMMENT  ;Mincpupct  = N/A
COMMENT  ;Maxcpupct  = N/A


*************************
COMMENT
COMMENT  Workgroup   = ES_Default
COMMENT  ;Memb_Queue = ES
COMMENT  ;Base       = 240
COMMENT  ;Limit      = 253
COMMENT  ;MinQuant   = 2000
COMMENT  ;MaxQuant   = 2000
COMMENT  ;Boost      = DECAY
COMMENT  ;Timeslice  = 2000
COMMENT  ;Mincpupct  = N/A
COMMENT  ;Maxcpupct  = N/A
```

The PROCS format displays the member processes and certain process attributes.

```
:SHOWWG Test_@; format= procs

 WORKGROUP    : Test_WG1
 PROCESSES    : 3


 PIN  PARENT    PRI   CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----  --------  -----   ------  --------------
 101  54       180 D  0:05.020  WAIT    S41     :tdp "text myfile"
%124  38       185 D  0:12.990  WAIT    J12     (TDP.PUB.SYS) text myfile
 189  67       183 D  0:01.030  READY   S57     (FCOPY.PUB.SYS)from=foo.pub.sys
 ;to=b$
********************

 WORKGROUP    : Test_WG2
 PROCESSES    : 1


 PIN  PARENT    PRI   CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----  --------  -----   ------  --------------
 173  32       240 D  0:08.120  WAIT    S71     :tdp "text report1"
********************
```

Note

The SHOWWG PROCS format is not presented as an atomic action. That is, the processes may be changing as they are being listed. Thus, you may see a listing where the number of processes at the beginning of the listing does not match the number of process line items. It is even possible for a given process to die, relinquishing its PIN to a new process. Thus, the same PIN might appear twice if the process dies after its first line item is listed.

The fields unique to the PROCS format are described below. The field's width, in characters, is shown within parentheses. A "v" indicates that the field has a variable size width.

WORKGROUP (v): The name of the specified workgroup. The workgroup name has a maximum length of 254 characters. However, the TRUNC option (default) will limit the name to the size of the user's $STDLIST. Truncation can be overridden with the NOTRUNC option. The "~" (tilde) as the first character of the workgroup name designates that it is in the purge-pending state.

PROCESSES (v): The number of processes within the specified workgroup.

PIN (6): Process Identification Number for the process. The format is [%]nnnnn, where the % (percent sign) indicates that the process is an artificial member of the workgroup. Processes are either natural or artificial members of a workgroup. Natural members meet the membership criteria, while artificial members have been placed explicitly into the workgroup (via ALTPROC or an AIF call).

PARENT (5): Process Identification Number for the process' parent (decimal). This field appears in the DETAIL and PROCS formats. These formats display PARENT to help you determine process relationships. Nothing is displayed when a process does not have a parent (for example, PROGEN).

PRI (5): The priority at which the process is currently executing. A lower numeric value indicates a higher priority. There is also an indication of whether the process is Linear (L) or Decayable (D). Linear processes run at fixed priority, whereas decayable processes experience priority decay.

CPUTIME (8): CPU is consumed in hh:mm:ss or m:ss.mls. A pair of asterisks (**) appears in the hours field when hours overflows. The three-character "mls" sub-field holds milliseconds.

STATE (5): This indicates the execution state of the process. STATE can be one of the following:

- BLKIO blocked for terminal write or control.
- WAIT generic process block, usually waiting for a message.
- BLKCB blocked for control block.
- BLKMM blocked for memory manager.
- READY ready to execute (or executing).

JOBNUM (6): The job or session number for the process.

PROGRAM (v): The file name of the program the process is executing.

STEP (v): The current command being executed by the CI process being displayed. This field is not shown for non-CI processes.

The DETAIL format displays the attributes associated with the specified workgroup(s). Below is a sample output created using the NOTRUNC option:

```
:SHOWWG Test_@; format= detail;notrunc

 WORKGROUP    : Test_WG1
 PROCESSES    : 3

        MEMBERSHIP CRITERIA
 PROGRAM  : @.@.TEST1
 QUEUE    : CS


             -----QUANTUM-----          TIME          CPU %
BASE LIMIT   MIN   MAX  ACTUAL  BOOST  SLICE PROCS  MIN MAX
---- -----  ----- ----- ------  -----  ----- -----  --- ---
180  200    1000  1000  1000    OSC    400   3      20  100



 PIN  PARENT    PRI   CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
-----  ------  -----  --------  -----   ------  --------------
 101  54      180 D  0:05.020  WAIT    S41     :tdp "text mytext"
%124  38      185 D  0:12.990  WAIT    J12     (TDP.PUB.SYS) text myfile
 189  67      183 D  0:01.030  READY   S57     (FCOPY.PUB.SYS)from=foo.pub.sys;to=ba
r;new
********************

 WORKGROUP    : Test_WG2
 PROCESSES    : 1


        MEMBERSHIP CRITERIA
 PROGRAM  : @.@.TEST2
 QUEUE    : CS, DS

             -----QUANTUM-----          TIME          CPU %
BASE LIMIT   MIN   MAX  ACTUAL  BOOST  SLICE PROCS  MIN MAX
---- -----  ----- ----- ------  -----  ----- -----  --- ---
170  180    2000   2000   2000  DECAY  600   1      30  40

 PIN  PARENT   PRI  CPUTIME   STATE   JOBNUM  (PROGRAM) STEP
------ ------ ----- --------  -----   ------  -------------
 173 140      240 D 0:08.120  WAIT    S71     :tdp "text test"
********************
```

Note

Regardless of format, SHOWWG lists workgroups as they appear in the ordered list, which is identical to the search order used to determine workgroup membership. SHOWWG gets this order and all other displayed values from system tables and not from files, thus showing the actual state of the system.

EXAMPLE

To display the active workgroups on an MPE/iX system, enter:

```
:SHOWWG @
```

To display the member processes of the CS_Default workgroup, enter:

```
:SHOWWG CS_Default; format= procs
```

To direct the current configuration of user-defined workgroups to a temporary file, and then save that file in the permanent file domain, enter:

```
:SHOWWG @; format= wgfile > file
:SAVE file
```

Related Information

Commands: NEWWG, ALTWG, PURGEWG, TUNE, SHOWQ, ALTPROC, SHOWPROC

Manuals: *MPE XL Intrinsics Reference Manual* (32650-90028)

TUNE

Changes scheduling characteristics of the scheduling subqueues. These characteristics include base and limit priorities, quantum bounds (min and max), boost property and timeslice. (Native Mode)

Syntax

$$\texttt{TUNE}\ [\ \textit{minclockcycle}\ ]\left\{\begin{array}{l}\texttt{;CQ=}\textit{qinfo}\\ \texttt{;DQ=}\textit{qinfo}\\ \texttt{;EQ=}\textit{qinfo}\end{array}\right\}[\ \ldots\ ]$$

Where *qinfo* is written in the following form:

$$\left[\ \textit{base}\ \left[\ \texttt{,}[\ \textit{limit}\ ]\left[\ \texttt{,}[\ \textit{min}\ ]\left[\ \texttt{,}[\ \textit{max}\ ]\left[\begin{array}{l}\texttt{,DECAY}\\ \texttt{,OSCILLATE}\end{array}\right]\texttt{,}[\ \textit{tslice}\ ]\ \right]\ \right]\ \right]\ \right]$$

Note

Misuse of this command can significantly degrade system operating efficiency.

PARAMETERS

minclockcycle **This parameter is ignored. It appears here for MPE V/E compatibility only.**

base An integer from 150 to 255 specifying the priority at which user processes executing in the CS, DS, and ES scheduling subqueues begin their Dispatcher transactions. Priority is inversely related to the integer: a higher-priority process has a lower number. While the full range is provided for compatibility, avoid setting the base priority between 150 and 152, since user processes running at priorities greater than 152 can adversely affect system performance.

limit An integer specifying the lowest priority at which a process in the CS, DS, or ES scheduling subqueues can execute. Priority is inversely related to the integer: a higher-priority process has a lower number. The *limit*, which can range from 150 to 255, must be greater than or equal to the *base*.

min The minimum quantum is a lower bound for the dynamically calculated quantum (average transaction time) value. The quantum value determines the rate of priority decay for processes within the scheduling subqueue. Values range between 1 and 32767 milliseconds.

max The maximum quantum is an upper bound for the dynamically calculated quantum (average transaction time) value. The quantum value determines the rate of priority decay for processes within the scheduling subqueue. Values range between 1 and 32767 milliseconds. The value of *max* must be greater than or equal to the value of *min*.

DECAY	Sets the subqueue to the default decay behavior associated with circular scheduling subqueues. If set, a process decays normally to the *limit* priority and returns to the *base* priority when the Dispatcher transaction is complete. DECAY is the default boost property.
OSCILLATE	Sets the subqueue to oscillate behavior. If set, a process returns to the *base* priority once its priority has decayed to the *limit* of the subqueue, even if it has not completed a Dispatcher transaction.
tslice	The timeslice is the number of milliseconds a process in a given subqueue can hold the CPU. A process that has held the CPU continuously for this number of milliseconds is interrupted. This value is accurate to the nearest increment of 100 milliseconds and has a minimum value of 100 milliseconds.

OPERATION

The system manager uses the TUNE command to change the characteristics of the circular scheduling subqueues to more efficiently manage the current processing load.

Processes in the CS, DS, and ES scheduling subqueues typically begin execution at the *base* priority. When a process stops (for disk I/O, terminal I/O, preemption, etc.), the amount of CPU it has consumed is used to determine its new priority. If the process has completed a Dispatcher transaction, typically by issuing a terminal read, its priority is reset to the *base*, and the quantum value for that workgroup is recalculated. If the process has exceeded the quantum (filter) value since its priority was last reduced, the priority is decreased without exceeding the *limit* priority. If the boost property for the workgroup is oscillate, process priorities are reset to the *base* value once they decay to the *limit*.

The parameters *min* and *max* refer to the absolute bounds of the quantum ("system average quantum" or SAQ), or a filter representing the average transaction time of processes in that subqueue. The quantum is recomputed after every user Dispatcher transaction is complete, and then compared against the CPU time of a process to determine whether the priority of the process should be decreased.

Note

With Release 5.0 of MPE/iX, all three circular scheduling subqueues, CS, DS, and ES, have dynamically calculated quantums. By default, the bounds of the DS and ES subqueues are set equal.

If the values specified for *max* are too large, system response may become erratic. If they are too small, excessive memory management may occur due to frequent process swapping. Either case degrades system performance. The values for *min* and *max* may range from 1 to 32,767.

The timeslice value determines how long a process in a given scheduling subqueue will be allowed to hold the CPU. This value is different than the quantum, which determines how rapidly process priorities decay. The timeslice does interrupt the process if the process is interruptable. The timeslice is a multiple of 100 milliseconds and has a minimum value of 100 milliseconds.

The following default settings are established when the system is booted from the system disk (a `START RECOVERY` or `START NORECOVERY`), unless the user has customized a `TUNE` configuration .

```
START RECOVERY or START NORECOVERY

CQ base:   152     DQ base:   202     EQ base:   240
  limit:   200       limit:   238       limit:   253
    min:     1         min:  2000         min:  2000
    max:  2000         max:  2000         max:  2000
  boost: DECAY       boost: DECAY       boost: DECAY
 tslice:   200      tslice:   200      tslice:   200
```

Note

The MPE/iX Scheduler now supports the workgroup concept. However, backward compatibility is maintained through five default workgroups created by the system. The scheduling characteristics of the CS_Default, DS_Default, and ES_Default workgroups mimic those of the CS, DS, and ES scheduling subqueues. In fact, changing the scheduling characteristics of the CS, DS, and ES scheduling subqueues, via the `TUNE` command, is equivalent to changing the characteristics of the corresponding default workgroup through `ALTWG`. Please refer to the `NEWWG` and `ALTWG` commands for more detail.

Workload Manager users should use `ALTWG` rather than `TUNE` since `TUNE` does not modify user-defined workgroups. If you aren't using the Workload Manager, and you want to change one of the system-defined workgroups, you may wish to use `ALTWG` because it only examines member processes of a specific workgroup and not all processes on the system.

The `TUNE` command may be issued from a session, job, program or in BREAK. Pressing Break has no effect on this command. `TUNE` requires System Supervisor (OP) or System Manager (SM) capability.

EXAMPLE

To set the CS subqueue's *base* to 152, *limit* to 200, and *max* quantum (filter) to 300; and the DS subqueue's *base* to 202, *limit* to 238, *min* and *max* quantum (filter) to 1000, and cause oscillation boosting, enter:

```
TUNE CQ=152,200,300,300;DQ=202,238,1000,1000,OSCILLATE
```

To set the CS subqueue to oscillation with a 300 millisecond timeslice and the DS subqueue's *base* to 180, *limit* to 238, boost property to decay, and timeslice to 1500, enter:

```
TUNE CQ=,,,,OSCILLATE,300;DQ=180,238,,,DECAY,1500
```

Related Information

Commands: `SHOWQ`, `ALTPROC`, `SHOWPROC`, `NEWWG`, `ALTWG`, `PURGEWG`, `SHOWWG`

Manuals: *MPE/iX Intrinsics Reference Manual* (32650-90028)

A

Troubleshooting

The Workload Manager is a powerful tool that can greatly impact system performance. Though it can be used to address performance issues, it can also be inadvertently misused, causing serious performance problems. This appendix includes information to help you troubleshoot problems that you may encounter using the Workload Manager.

Troubleshooting Workgroup Problems

Each of the following sections lists a potential problem and a list of possible solutions.

Uncaptured process

Problem: You have created user-defined workgroups but a process does not fall into the expected workgroup.

Solutions:

system process
: The process might be a system process, which cannot be captured by user-defined workgroups. System processes are always members of one of the system-defined default workgroups. The `SHOWPROC` command displays an asterisk ("`*`") next to the pin number of system processes.

artificial member
: The process might be an artificial member of its workgroup. An artificial member remains in that workgroup until the workgroup is deleted, the process is moved to another workgroup, or the process is returned to its natural workgroup. The DETAIL format of the `SHOWPROC` command places a percent sign ("`%`") next to the workgroup name if the process is an artificial member. Also, a percent sign ("`%`") appears before the pin number on the `DETAIL` format of the `SHOWPROC` command or the `PROCS` format of `SHOWWG` command. To return a process to its natural workgroup, issue the command `ALTPROC` *pin*`;WG=Natural_wg`.

membership criteria
: The process might not meet the membership criteria for the specified workgroup. The `DETAIL` format of the `SHOWPROC` command displays the process attributes on which membership can be based (logon, program, queue) and the workgroup of the process. The `DETAIL` format of the `SHOWWG` command displays the membership criteria of the workgroup. To be a member, the process must match *one* value of *each* of the specified categories. If the process doesn't match, it will not be a member. If it does match, it *might* be a member of the workgroup, depending on its position in the ordered list of workgroups.

workgroup order
: Workgroup membership is determined by scanning the workgroup membership criteria of all workgroups in order. The three process attributes on which membership can be based (logon, program, queue) are compared to the workgroup membership criteria, and the process is placed in the first matching workgroup. Due to this, you place workgroups with the most specific membership at the beginning of the list, and place workgroups with more general

membership criteria near the end. The workgroup with the *most* general membership criteria are the five system-defined workgroups, which always appear last.

To see the current order, which is the order workgroups appear in the workgroup configuration file, issue the `SHOWWG` command . (You can control this order by introducing a new workgroup configuration, or by specifying the `POSITION` parameter on the `NEWWG` command line when you create a new workgroup.)

Starving workgroup

Problem: All processes within a workgroup are not receiving sufficient CPU time.

Solutions:

low priority

The priority range assigned to the workgroup might be low when compared to the priority ranges of other workgroups. Use the `SHOWWG` command to display the base and limit priorities of the various workgroups. If the starving workgroup is at low priority when compared to the other workgroups, moving it to a higher priority range would help it get more CPU. Or, you can determine which workgroup(s) are impacting the affected workgroup and change them.

minimum(s) too low

Minimum CPU values provide a guarantee that the workgroup will receive the specified amount of CPU, if the workgroup requires it. There are two situations in which CPU minimums can starve a workgroup:

- A minimum CPU value has been assigned to the starving workgroup that is not sufficient to meet the demand of processes within that workgroup. In this situation, raise the CPU minimum for that workgroup, making sure that the cumulative minimums for all workgroups is less than 100%.
- The other workgroups have been assigned (and use) CPU minimum values that do not leave sufficient CPU for the starving workgroup. In this case, lower the CPU minimums of one or more workgroups to allow sufficient CPU for the starving workgroup.

maximum too low
: The workgroup may also be starving because it has been assigned a maximum amount of CPU that is insufficient. For example, the workgroup may be assigned a maximum CPU value of 20% and be using that 20%, but requires 30% for adequate response time. Alternatively, other workgroups that should be constrained by their maximums might have those values set too high. A workgroup with a maximum of 80% would be allowed to consume 80% of the system, provided it did not violate any minimum guarantees for other workgroups.

Starving process

Problem: A workgroup might be receiving its share of the CPU, but a process within that workgroup is not getting sufficient CPU time.

Solutions:

check priority
: The MPE/iX Dispatcher remains priority-driven, allocating the CPU to the processes within the workgroup based on their priority. Use the `PROCS` format of the `SHOWWG` command to display the priorities of member processes of the workgroup(s) you specify. If the process is of lower priority than other member processes, it will only receive the CPU that the other processes do not require.

change workgroup
: The process might belong in a workgroup with higher priority values. For example, a batch job might be particularly important and deserve to run at higher priority than most batch jobs. Use the `ALTPROC` command to move the process to another workgroup.

  Note: In the interests of being proactive, you might want to define a workgroup with membership criteria that would naturally capture that process, placing it at an appropriate priority.

enable oscillation
: The process may be in the proper workgroup, but is having trouble competing with other processes in the workgroup since its transaction time is greater than the average. Recall that the priority of a process will decay based on its CPU consumption. The process may have decayed to the limit of the queue and is unable to compete for the CPU with other processes of higher priority in that workgroup. Use the `ALTWG` command to enable oscillation, which will boost the priority of any process that decays to the limit of the workgroup.

adjust quantum bounds	If the process is at lower priority than other processes in the workgroup, but hasn't decayed to the limit priority (so that oscillation will take place), the rate of priority decay can be changed. Use the `ALTWG` command to change the quantum bounds to reduce the quantum. A smaller quantum ensures faster priority decay so that processes decay to the limit more quickly and can be oscillated.

CPU minimum not met

Problem: The observed CPU allocation to a workgroup is less than the minimum CPU percentage.

Solutions:

insufficient demand	If the processes within the workgroup do not require the amount of CPU they have been guaranteed, then the observed CPU allocation will be lower than the set value. If the processes can consume only 20% of the CPU, but have been given a minimum of 30%, they will not be able to consume the minimum amount.
too few processes	If you are using a system with multiple processors and there are a small number of processes running, the CPU demand may not reach the minimum assigned to the workgroup. For example, suppose your system has four CPUs and you have assigned a 40% minimum CPU percentage to a workgroup. If there is only one process running in that workgroup, the workgroup can consume a maximum of 25% of the total CPU capacity of the system, or one processor.

System or process hang

Problem: You have set the maximum CPU percentage of a workgroup to zero and it has starved. Or, a workgroup that captures processes running `CI.PUB.SYS` has no CPU access (either by setting the CPU maximum to zero or by placing them at low priority on a busy system). Or, the CPU minimum percentage guarantees do not allow sufficient CPU for processes in the default workgroups.

Solutions:

if you can enter commands	Adjust the scheduling characteristics of the problem workgroups or delete the entire workgroup configuration. If you are able to identify the problem workgroup(s) (e.g., a workgroup with a maximum CPU % of zero), use the `ALTWG` command to alter the scheduling characteristics. If you are uncertain of the problem workgroup(s) and wish to remove all user-defined workgroups, leaving only the five system-defined default workgroups, issue the command `PURGEWG @`.

if you can't enter commands	Reboot the system and at the `ISL` prompt, enter the command `START SINGLE-USER`. This initiates single-user mode, in which only the five default workgroups are available; all user-defined workgroups are purged. Next, use the `NEWWG` command to invoke a more appropriate workgroup configuration (or you can choose to stay with just the five default workgroups). Finally, enter the command `START` or `START MULTI-USER` to bring the system up multi-user.

B

Error Messages

This appendix describes error messages returned by the Command Interpreter (CI) that relate to workgroups and using the Workload Manager. For each message, you will see the cause or probable cause and one or more suggestions to remedy the error.

11506	MESSAGE	SHOWPROC format must be SUMMARY or DETAIL. (CIWARN 11506)
	CAUSE	You entered a format name other than the two supported by the SHOWPROC command, SUMMARY or DETAIL. Or, you misspelled one of the two format names.
	ACTION	Reenter the command and specify either SUMMARY or DETAIL as the format name.
12201	MESSAGE	Workgroup name already exists. (CIERR 12201)
	CAUSE	The name that you specified for the new workgroup already exists.
	ACTION	Use the SHOWWG command to determine whether or not a workgroup by that name exists. If not, check the spelling of the workgroup name and reenter the command.
12202	MESSAGE	Workgroup does not exist. (CIERR 12202)
	CAUSE	There is no workgroup whose name matches the name that you specified.
	ACTION	Use the SHOWWG command to determine whether or not a workgroup by that name exists. If not, check the spelling of the name that you specified and reenter the command.
12203	MESSAGE	Cannot delete a system-defined workgroup. (CIERR 12203)
	CAUSE	You attempted to delete one of the following workgroups, which are defined by the system and cannot be purged: AS_Default, BS_Default, CS_Default, DS_Default, and ES_Default.
	ACTION	If you are using the Workload Manager, reenter the command and specify the name of the user-defined workgroup that you want to delete. If you are not using the Workload Manager, you will always see this error message when you issue the PURGEWG command since you cannot create user-defined workgroups. Regardless of whether or not you are using the Workload Manager, you cannot create or delete the system-defined workgroups.
12204	MESSAGE	LIMIT priority cannot exceed BASE. (CIERR 12204)
	CAUSE	The value that you specified for the limit priority exceeds the value of the base priority.
	ACTION	Reenter the command to supply a value for *limit* that is less than or equal to the value for *base*. If you do not know the current value for *base*, use the SHOWWG command to display that information. Remember that the value of *limit* must be between 150 and 255, inclusive.

12205	MESSAGE	`MINQUANT and MAXQUANT values must be between 1 and 32767, inclusive. (CIERR 12205)`
	CAUSE	The value that you specified for the minimum quantum or the maximum quantum is not within the allowable limits of 1 and 32767, inclusive.
	ACTION	Reenter the command, specifying a value between 1 and 32767 inclusive. Remember that the minimum quantum value must be less than or equal to the maximum quantum.
12206	MESSAGE	`MINQUANT value cannot exceed MAXQUANT. (CIERR 12206)`
	CAUSE	The value that you supplied for the minimum quantum exceeds the value of the maximum quantum.
	ACTION	Reenter the command supplying a minimum quantum value that is less than or equal to the maximum quantum. If you do not know the current value of *maxquant*, use the `SHOWWG` command to display that information.
CIERR 12207	MESSAGE	`TIMESLICE value must be a multiple of 100 and within the range 100 and 32700, inclusive. (CIERR 12207)`
	CAUSE	The value that you specified for the timeslice is either not a multiple of 100 or it is not within the range of 100 and 32700, inclusive.
	ACTION	Reenter the command to specify a timeslice value that is a multiple of 100 (i.e. 200, 300, 400, ... 15100, 31900). The smallest acceptable value is 100 and the largest acceptable value is 32700.
CIERR 12208	MESSAGE	`Total minimum CPU percentages for all user-defined workgroups cannot exceed 99. (CIERR 12208)`
	CAUSE	The total of all the minimum CPU percentages specified for all the user-defined workgroups exceeds 99.
	ACTION	Calculate the sum of CPU percentages for all existing user-defined workgroups and subtract the amount from 99. Reenter the command specifying a minimum CPU percentage that is equal to or less than this amount. If the sum of the minimum CPU percentages of the existing workgroups is already equal 99, you cannot specify a minimum CPU percentage for the workgroup that you are trying to create.
CIERR 12209	MESSAGE	`MAXCPUPCT value must be within 0 and 100, inclusive. (CIERR 12209)`
	CAUSE	The value that you specified for maximum CPU percentage is either less than zero, or it is greater than 100.
	ACTION	Reenter the command to specify a valid percentage value. Make sure that *maxcpupct* is not less than *mincpupct*.

CIERR 12210	MESSAGE	MINCPUPCT value must be within 0 and 99, inclusive. (CIERR 12210)
	CAUSE	The value that you specified for minimum CPU percentage is either less than zero, or it is greater than 99.
	ACTION	Reenter the command to specify a valid percentage value. Make sure that *mincpupct* does not exceed *maxcpupct*, and that the sum of all minimum CPU percentages is less than 100.
CIERR 12211	MESSAGE	MINCPUPCT value cannot exceed MAXCPUPCT. (CIERR 12211)
	CAUSE	The value that you specified for the minimum CPU percentage is greater than the maximum CPU percentage.
	ACTION	Reenter the command to specify a minimum CPU percentage that is less than or equal to the maximum. If you do not know the current value of *maxcpu*, use the SHOWWG command to display that information.
12212	MESSAGE	Workgroup name must start with a letter or underscore and can contain 255 alphanumeric and underscore characters. (CIERR 12212)
	CAUSE	You entered a workgroup name that is longer than 255 characters or you entered an invalid first character.
	ACTION	Make sure that the first character of the name is either a letter or an underscore; numbers and special characters are not permitted. Also, check the length of the workgroup name to ensure that it is less than or equal to 255 characters. (The name may have unprintable characters that are not visible on the screen but are included in and contribute to the length of the input. They are saved as part of the command in the CI history stack; use the DO or REDO commands to check.) Once you have determined the cause of the problem, reenter the command to specify the corrected workgroup name.
12213	MESSAGE	Invalid character in workgroup name. (CIERR 12213)
	CAUSE	You entered a workgroup name that begins with a character other than the normal alphabet letters or underscore (_), or one that contains a character other than a number, letter, or the underscore.
	ACTION	Check the name for illegal or unprintable characters. (Unprintable characters are control characters that are not visible on the screen but are included in the input. They are saved as part of the command in the CI history stack use the DO or REDO commands to check.) Once you have determined the cause of the problem, reenter the command to specify the corrected workgroup name.
12214	MESSAGE	SHOWWG format must be 0, 1, 2, 3, or format name. Defaulting to format SUMMARY (0). (CIWARN 12214)
	CAUSE	You entered a format number that is not between 0 and 3 inclusively.
	ACTION	Reenter the command to specify one of 0, 1, 2, 3. Or, reenter the command to specify one of the following format names: SUMMARY, WGFILE, PROCS, or DETAIL.

12215	MESSAGE	NOTRUNC option is ignored for WGFILE format. (CIWARN 12215)
	CAUSE	You attempted to display workgroup information in the WGFILE (1) format and specified the NOTRUNC option.
	ACTION	Reenter the command to omit the NOTRUNC option. Or, choose either the PROCS or DETAIL format to display workgroup information without truncation and specify the NOTRUNC option.
12217	MESSAGE	VALIDATE option is ignored when not specified in conjunction with an indirect file name. (CIWARN 12217)
	CAUSE	You specified the VALIDATE option on the command line to create or change a single workgroup. Or, you did not enter the caret (^) preceding the indirect file name and the CI interpreted it as a workgroup name, thereby disallowing the VALIDATE option.
	ACTION	Enter the command again, making sure that a caret (^) immediately precedes the indirect file name. (If you do not specify this character, the file name will be treated as a workgroup name.)
12218	MESSAGE	Only VALIDATE option is allowed with indirect file. Other parameters are ignored. (CIWARN 12218)
	CAUSE	You specified the name of an indirect file on the command line but supplied options other than VALIDATE. Or, you preceded the workgroup name with the caret (^), and the CI interpreted it as the name of an indirect file.
	ACTION	Check the command line to ensure that you did not inadvertently enter a caret (^) with the workgroup name. Or, if you meant to supply the name of an indirect file, reenter the command and only specify the VALIDATE option with that name.
12219	MESSAGE	At least one membership criterion is required. (CIERR 12219)
	CAUSE	You attempted to create a workgroup but did not specify a membership criterion.
	ACTION	Reenter the command to supply at least one of the following membership criteria: MEMB_LOGON, MEMB_PROGRAM, or MEMB_QUEUE.
12220	MESSAGE	POSITION parameter is not allowed within an indirect file. (CIERR 12220)
	CAUSE	One or more of the workgroup specifications in the indirect file that you supplied to the NEWWG command contain the POSITION parameter.
	ACTION	Edit the indirect file to remove all the POSITION parameter specifications from the workgroup specifications. Then, reenter the NEWWG command.

12221	MESSAGE	VALIDATE option is not allowed within an indirect file. (CIERR 12221)
	CAUSE	One or more of the workgroup specifications in the indirect file that you supplied to the NEWWG command contain the VALIDATE option.
	ACTION	Edit the indirect file to remove all of the VALIDATE option entries from the workgroup specifications. Then, reenter the NEWWG command.
12222	MESSAGE	The indirect file may not contain references to itself or to another indirect file. (CIERR 12222)
	CAUSE	One or more of the workgroup specifications in the indirect file that you supplied to the NEWWG command contains a name that the CI is interpreting as indirect file instead of a workgroup.
	ACTION	Edit the indirect file to remove all caret (^) characters that might cause a workgroup name to be interpreted as an indirect file name. Then, reenter the NEWWG command.
12223	MESSAGE	BASE and LIMIT priorities must be specified. (CIERR 12223)
	CAUSE	A workgroup specification in the indirect file does not include values for the base priority or the limit priority, which are required.
	ACTION	Edit the indirect file to add a value for *base* and a value for *limit* to the workgroup specification whose name appears in the text of the error message.
12224	MESSAGE	MPE/iX names must be fully qualified, and HFS names must be absolute paths. (CIERR 12224)
	CAUSE	When specifying the MEMB_PROGRAM values, you did not fully qualify an MPE/iX file name or you did not enter an absolute path for an HFS name.
	ACTION	Reenter the command to ensure that the MPE/iX file names contain both the group and account components, and that HFS names begin with a / (denoting an absolute path.) Remember that HFS paths which reference MPE/iX file names must be in all capital letters (e.g. /SYS/PUB/CI).
12225	MESSAGE	No workgroup names matched the pattern "!". (CIWARN 12225)
	CAUSE	None of the existing workgroup names match the pattern that you specified.
	ACTION	Check the order of the wildcards in the pattern to ensure that it indicates what you intended and then reenter the command to specify the correct wildcards. Remember that "#" means one numeric character, "?" means one alphanumeric character, and "@" means one or more alphanumeric characters.

CIERR 12226	MESSAGE	PRI and WG parameters may not be specified together. (CIERR 12226)
	CAUSE	You specified both PRI and WG parameters, and the ALTPROC command can accept only one of them.
	ACTION	Choose which parameter you want to use and reenter the command to specify only that one. Use the WG parameter to explicitly place a process in a workgroup (where it becomes an *artificial member*) or to return a process to its natural workgroup. Use the PRI parameter to alter the priority of a process. Note: Workload Manager users should avoid using the PRI parameter, and use WG instead.
12227	MESSAGE	WG parameter requires SM capability. (CIERR 12227)
	CAUSE	You cannot use the WG parameter to change the workgroup of the specified process because you have not been assigned system manager (SM) capability.
	ACTION	Consult your system manager or system administrator.
12228	MESSAGE	Command does not allow wildcards in workgroup name. (CIERR 12228)
	CAUSE	You entered wildcard characters in the workgroup name specified with the NEWWG or ALTWG commands, or with the WG= parameter of ALTPROC command.
	ACTION	Remove any wildcards from the workgroup name and reenter the command.
12229	MESSAGE	POSITION parameter does not specify a valid workgroup name. (CIERR 12229)
	CAUSE	The workgroup name that is specified in the POSITION parameter does not exist, or it is one of the following system-defined workgroups: BS_Default, CS_Default, DS_Default, or ES_Default.
	ACTION	Check the spelling of the workgroup name to ensure that it is what you intended. If necessary, you can display a list of existing workgroup names by entering the SHOWWG command. Also, since you cannot place a user-defined workgroup after the system-defined workgroups, the POSITION parameter does not accept any of the following workgroup names: BS_Default, CS_Default, DS_Default, or ES_Default. (You may specify the AS_Default workgroup, however, because the workgroup you are adding will precede the workgroup you specify in the POSITION parameter. So, for example, you would specify the AS_Default workgroup when you want the workgroup you are adding to be *the last user-defined workgroup* in the ordered list.)

12230	MESSAGE	`This name is reserved for a system-defined workgroup. (CIERR 12230)`
	CAUSE	You attempted to create a user-defined workgroup using one of the system-defined workgroup names.
	ACTION	Reenter the command to specify a workgroup name that does not match one of the system-defined workgroups: AS_Default, BS_Default, CS_Default, DS_Default, and ES_Default or the reserved workgroup name Natural_wg. Although these workgroup names appear in mixed alphabetic case, the system reserves these names irrespective of case. So if you attempt to create a new workgroup called `es_default`, for example, you will see this error message.
12231	MESSAGE	`Cannot modify the AS_Default and BS_Default workgroups. (CIERR 12231)`
	CAUSE	You attempted to redefine one or more characteristics of the AS_Default or the BS_Default system-defined workgroups.
	ACTION	You cannot alter the characteristics of the AS_Default and BS_Default workgroups. Make sure that the workgroup name is what you intended and reenter the command to specify the correct workgroup name.
12235	MESSAGE	`The Workload Manager configuration file is corrupt. Enter "HELP CIERR12235" for more information. (CIERR 12235)`
	CAUSE	For some reason, the `WGCONFIG` file has been corrupted.
	ACTION	Reboot the system and, at the ISL prompt, enter `START NORECOVERY SINGLE-USER`. Rebooting the system in this way renames the corrupted `WGCONFIG` file and creates a new `WGCONFIG` file with the system-defined (default) workgroups. Once the system is finished booting, you can use the `NEWWG` command to add user-defined workgroups to the configuration. Once you have done so, restart the system with your normal `START` command specifications.
12236	MESSAGE	`Indirect file must be an ASCII file. You may use the WGFILE format of the NEWWG command in conjunction with I/O redirection to create one. (CIERR 12236)`
	CAUSE	The indirect file you specified on the command line is not an ASCII file.
	ACTION	Use an editor such as EDIT/3000 to create an ASCII file that contains the definition(s) of your user-defined workgroup(s). You can either create this file from scratch or, if you intend to modify the current configuration, you can capture the output of the SHOWWG command and edit that. To do so, enter `SHOWWG @, WGFILE >` *filename*. This creates a temporary ASCII file, of the name you specified, that contains the current definitions of all the workgroups that are active on your system.

12237	MESSAGE	`Cannot rewind the indirect file for reading. (CIERR 12237)`
	CAUSE	An access error occured during repositioning of the record pointer to the beginning of the indirect file.
	ACTION	Examine the file to ensure that it is not corrupt, then reenter the command.
12238	MESSAGE	`You must purchase the Workload Manager (product # B3879AA) before you can create any user-defined workgroups. (CIERR 12238)`
	CAUSE	You tried to create a user-defined workgroup without purchasing the Workload Manager. If you have purchased the product and you see this message, it is possible that one or more of its components have been corrupted.
	ACTION	If you have purchased the Workload Manager, verify that the `WLMGR.MPEXL.SYS` file is intact. If it is not, follow the steps in the installation guide to regenerate it. To purchase a copy of the Workload Manager, contact your local Hewlett-Packard sales office.
12239	MESSAGE	`Logon name must be of the form "[JobSess_Name,]USER.ACCOUNT" where wildcards are allowed. (CIERR 12239)`
	CAUSE	The `MEMB_LOGON` parameter contains a logon name that does not follow the requirements of the Workload Manager logon name specification.
	ACTION	Inspect the logon name to ensure that each component is not more than eight characters long and that it begins with an alphabetic character or either the "`@`" or "`?`" wildcard characters. (Within the logon name, the wildcard characters "`@`", "`?`" and "`#`" are permitted. However, since names may not begin with digits, you cannot use "`#`" as the first character of any of the logon components.) Also, if the `JobSess_Name` portion of the logon is supplied, the whole name must be quoted so that it is treated as one entity. Otherwise, the comma separating `JobSess_Name` from `USER.ACCOUNT` will cause `JobSess_Name` to be treated as a logon name by itself.

12240	MESSAGE	`Cannot change the default workgroup of a system process. (CIERR 12240)`
	CAUSE	You attempted to change the workgroup of a system process, which is not allowed.
	ACTION	Use the `SHOWPROC` command to make sure that the process whose workgroup you are changing is a user process. If it is, verify that the PIN number that you supplied is correct and then reenter the command.
12241	MESSAGE	`Cannot modify the CPU percentages of the system-defined workgroups. (CIERR 12241)`
	CAUSE	You attempted to change the CPU percentages of one of the AS_Default, BS_Default, CS_Default, DS_Default, or ES_Default system-defined workgroups. You can only change the CPU percentages of the user-defined workgroups.
	ACTION	Check the spelling of the workgroup name that you specified, keeping in mind that names are case insensitive. Reenter the command to supply a user-defined workgroup name.

Glossary

artificial member
A process that has been explicitly placed in a workgroup via the `:ALTPROC` command or `AIFPROCPUT`, instead of naturally meeting the membership criteria of the workgroup.

circular scheduling subqueues
A name for the CS, DS, and ES scheduling subqueues on systems running version 4.7 or earlier of MPE/iX. These subqueues are considered circular because the priority of processes decays over time, circulating within the bounds of the established base and limit values for the subqueue. For example, a process in the CS subqueue will start at the base priority of 152 and decay towards the limit of 200. At some point, depending upon the boost property set for the subqueue, the process is boosted back to the base priority of 152 to continue execution.

CPU scheduling
The access a process has to the CPU. The MPE/iX Scheduler allocates CPU time to processes based on their priority. The system manager can control a process' priority (and hence, how much of the CPU it receives relative to other processes) via the creation of workgroups.

process
A program currently being executed.

purge-pending workgroups
A workgroup that the system manager has purged (via the `PURGEWG` command) that still has member processes which have not yet died or migrated to another workgroup.

purgescan
A review of all processes (scan) that occurs after the system manager issues the `PURGEWG` command.

stream
To submit a job for processing, either to begin now or at some specified time in the future.

system-defined workgroups
The five default workgroups present on all systems running Release 5.0 of MPE/iX: AS_Default, BS_Default, CS_Default, DS_Default, and ES_Default. Each system-defined workgroup corresponds to a traditional scheduling subqueue. The AS_Default workgroup, for example, corresponds to the AS subqueue.

system-wide scan
A review of all system and user processes to determine their appropriate workgroup membership. A system-wide scan occurs after the system manager alters the existing workgroup configuration, for example, after purging a workgroup.

traditional scheduling subqueues
The five scheduling subqueues available on systems running Release 4.7 or earlier of MPE/iX. The AS and BS scheduling subqueues are used for non-decayable (i.e. linear) system processes, while the remaining three subqueues, CS, DS, and ES, are used for decayable (i.e. circularly-scheduled) user processes. On MPE/iX 5.0 systems, the traditional scheduling subqueues are replaced by five system-defined workgroups.

tune
To modify the scheduling characteristics of a subqueue or workgroup. System managers use the `:NEWWG` command to define scheduling characteristics for new workgroups that they are creating. Subsequently, they can use the `:ALTWG` command to change (or tune) these characteristics. Or, they can use the `:TUNE` command (from which the expression "to tune" is derived) to modify the CS, DS, and ES subqueues; that is, the CS_Default, DS_Default, and ES_Default workgroups.

user-defined workgroups
A workgroup that the system manager creates using the `:NEWWG` command. All user-defined workgroups have a unique workgroup name, a set of membership criteria (which must include at least one of `MEMB_LOGON`, `MEMB_PROGRAM`, or `MEMB_QUEUE`), and assigned scheduling characteristics (such as *base* and *limit* priority, *timeslice*, CPU percentage bounds, and so on). The system manager creates user-defined workgroups to partition the system workload for greater control over CPU scheduling.

workgroups
Entities that dictate the behavior of the user processes that belong to them. Membership in a workgroup is dynamic and can be determined by a number of factors, including the logon, program, and scheduling subqueue attribute of the process. On all systems, there are five system-defined workgroups. Using the Workload Manager, system managers can create user-defined workgroups.

Index

D