

LEVEL 68  
MULTICS RESOURCE CONTROL  
USERS' GUIDE

**SUBJECT**

Overview of the Resource Control Package/Resource Management Facility,  
Including A Description of User Commands and Procedures

**SOFTWARE SUPPORTED**

Multics Software Release 9.0

**ORDER NUMBER**

CT38-00

June 1981

**Honeywell**

## PREFACE

The purpose of this manual is to provide a detailed overview of the Multics Resource Management facility, including the Resource Control Package, and to explain why and how it is used.

The Multics Administrators' Manual-Resource Control, Order No. CC74 is the primary Resource Management Facility document, which explains to administrators the details of running the facility, including maintenance and restoration of the registries.

Non-administrative users of the facility can gain sufficient understanding from the material contained in this manual. Administrative users additionally require MAM Resource Control to perform their duties of setting up and maintaining the environment.

The reference manuals for Multics administrators are collectively referred to as the Multics Administrators' Manual (MAM). Throughout this document, references to the MAM are as follows:

<u>Document</u>	<u>Referenced In Text As</u>
<u>Project</u> (Order No. AK51)	MAM Project
<u>Registration and Accounting</u> (Order No. AS68)	MAM Accounting
<u>System</u> (Order No. AK50)	MAM System
<u>Resource Control</u> (Order No. CC74)	MAM RCP
<u>Communications</u> (Order No. CC75)	MAM Communications

The MAM Project is a guide to the operation of programs in the project-administration area. The information in this manual is of interest not only to project administrators but also to accounting administrators (who may function as project administrators) and to system administrators (who may function in any administrative capacity).

The information and specifications in this document are subject to change without notice. This document contains information about Honeywell products or services that may not be available outside the United States. Consult your Honeywell Marketing Representative.

The MAM Accounting is a guide to the operation of Multics billing and accounting programs. It is necessary that both the accounting and system administrators know how to perform the Multics billing operations.

The MAM System is a guide to the overall administration of the Multics system. This manual discusses the contents of administrative directories and data bases and special user identities (such as the daemons), describes installation parameters and system logs, explains the various tasks that are the responsibility of the system administrator, and includes the commands needed to carry out these responsibilities. Also, the functions of the system security administrator are explained in the MAM System.

The MAM RCP is a guide to the management of the use of peripheral I/O devices and the physical volumes which can be mounted on these devices. This manual describes registering and deregistering such resources, controlling the access to these resources, and the pertinent administrative commands.

The MAM Communications is a guide to the operation of the Multics Communication System (MCS). The manual includes information on terminal types, line types, and channel management.

The primary reference manuals for user and system programmers of the Multics system are collectively referred to as the Multics Programmers' Manual (MPM). These manuals contain general information and programs needed by an administrator and are, therefore, referenced throughout this document. For convenience, these references are as follows:

<u>Document</u>	<u>Referenced In Text As</u>
<u>Reference Guide</u> (Order No. AG91)	MPM Reference Guide
<u>Commands and Active Functions</u> (Order No. AG92)	MPM Commands
<u>Subroutines</u> (Order No. AG93)	MPM Subroutines
<u>Subsystem Writers' Guide</u> (Order No. AK92)	MPM Subsystem Writers' Guide
<u>Peripheral Input/Output</u> (Order No. AX49)	MPM Peripheral I/O
<u>Communications Input/Output</u> (Order No. CC92)	MPM Communications I/O

CONTENTS

		Page
Section 1	Introduction . . . . .	1-1
	Overview of the Resource Management Facility	1-1
	Reservation, Assignment, and Attachment .	1-2
	Summary of Resource Management Facility	
	Actions . . . . .	1-4
	Acquisition . . . . .	1-4
	Resource Reservation . . . . .	1-4
	Device Assignment . . . . .	1-5
	Device Attachment . . . . .	1-6
	Resource Naming Conventions . . . . .	1-6
	Device Names . . . . .	1-6
	Volume Names . . . . .	1-7
	Access Control . . . . .	1-7
	Access Control Segments . . . . .	1-7
	Access Class Ranges . . . . .	1-8
	RCP Effective Access . . . . .	1-8
	Manipulating RCP Effective Access . . . .	1-9
	Acquiring Resources . . . . .	1-10
	Acquisition to the System Pool . . . . .	1-10
	Acquisition by User . . . . .	1-10
	Acquisition Upon Registration . . . . .	1-10
	Sites Not Enabling Resource Management	1-10
	Device Limits . . . . .	1-10
	I/O Workspaces . . . . .	1-11
	Glossary . . . . .	1-12
Section 2	How To Use Resource Management . . . . .	2-1
	User's Guide to Tapes at Sample Site . . . .	2-1
	Tape Registration . . . . .	2-1
	Acquisition Of Tapes . . . . .	2-1
	Access Control . . . . .	2-2
Section 3	Administrative and User Interfaces . . . . .	3-1
	Administrative Data Bases and Interfaces . .	3-1
	User Interfaces . . . . .	3-1
	Acquiring and Using Resources . . . . .	3-1
	Naming Rules for Attributes . . . . .	3-2
	User Manipulation of Registries . . . . .	3-2
Section 4	User Commands . . . . .	4-1
	acquire_resource, aqr . . . . .	4-4
	assign_resource, ar . . . . .	4-7
	cancel_resource, cnr . . . . .	4-10
	list_resource_types, lrt . . . . .	4-11
	list_resources, lr . . . . .	4-12
	release_resource, rlr . . . . .	4-15
	reserve_resource, rsr . . . . .	4-16
	resource_status, rst . . . . .	4-18
	set_resource, setr . . . . .	4-20
	unassign_resource, ur . . . . .	4-22

CONTENTS (cont)

	Page
Section 5	
User Subroutines . . . . .	5-1
cv_rcp_attributes_ . . . . .	5-3
resource_control_ . . . . .	5-9
resource_info_ . . . . .	5-16
Index . . . . .	i-1

TABLES

Table 1-1.	RCP Effective Access . . . . .	1-9
Table 1-2.	I/O Workspaces . . . . .	1-11

## SECTION 1

### INTRODUCTION

This manual contains the information necessary to understand the Resource Management Facility from a user's point of view and to acquire, reserve, and use resources as well as releasing them.

Section 2 describes how to use Resource Management, using as an example tapes at a sample site. Section 3 describes the user interfaces in detail and gives an overview of administrative interfaces. Section 4 contains descriptions of user commands; Section 5 contains the user subroutines.

### OVERVIEW OF THE RESOURCE MANAGEMENT FACILITY

The resource control package (RCP) resource management facility is the part of the Multics operating system that manages the use of peripheral I/O devices (such as tape drives, printers, and punches) and physical volumes that can be mounted on these devices (such as tape reels, forms, and disk packs). These resources are managed by programs located in the administrative ring (ring 1), and run in the user's process.

The resource management facility handles registration and acquisition of resources, which includes deregistration and release.

RCP software reserves, assigns, and mounts resources; also demounts, unassigns, and cancels reservations.

The hierarchical level of these functions are:

1	register	}	Resource Management
2	acquire		
3	reserve	}	Resource Control
4	assign		
5	attach		
5	detach		
4	unassign		
3	cancel	}	Resource Management
2	release		
1	deregister		

The function of RCP is to control the access to and usage of I/O devices. RCP executes in ring 1. Access to the various functions of RCP are controlled by the ring 1 gates that must be used to call RCP. One of the primary functions of RCP as a device manager is to control access to the I/O interfacier (IOI). In order to do this, no IOI gate entries are available to perform device attachments, detachments, and other privileged administrative functions. User ring programs, therefore, call RCP in order to request IOI to perform these functions.

An important part of RCP is Resource Management--the ability to retain registration information for all resources that it controls. It does this by providing administrative interfaces for the registration of resources (see Section 3). Registration of a resource provides information such as: what type of resource this is, what its name is, which attributes it possesses, or in what access class range the resource can be used. Once a resource is registered, users may acquire it; system administrators can also acquire it to a user (or to the system pool) at the time it is registered (described in the MAM RCP). The act of acquisition makes a user the owner of the resource--liable for all charges to that resource and in control of discretionary access to the resource.

Another important feature of RCP is its ability to control access to the various resources that it manages (where a resource is either a device or a volume). It does this through the use of access control segments (ACSs). An ACS is a zero-length segment whose ACL and ring brackets are used to define the discretionary and intraprocess access to a resource. At a site's discretion, additional features of RCP can be enabled to provide nondiscretionary access control for resources. If this is done, access is also controlled by the AIM access class range of a resource (see "Access Control" below).

The resource control functions performed by RCP are:

1. maintain resource information
2. control access to resources
3. reserve and cancel reservation of resources
4. assign and unassign devices
5. attach and detach devices
6. perform special device control functions

### Reservation, Assignment, and Attachment

The functions reserve, assign, and attach are organized into hierarchical levels. Defaults are provided at each level so that users not desiring to exercise features specific to a level do not have to concern themselves with that level.

- 1 reserve
- 2 assign
- 3 attach
- 3 detach
- 2 unassign
- 1 cancel

The first level involves the reservation of resources by processes. Tape drives, disk drives, tape volumes and disk volumes can be reserved. Reservations are process-specific and remain in effect until the process requests a cancellation. Reservation implies that a process temporarily has exclusive rights to a resource. This exclusive right means that no other process can use that resource for the duration of the reservation. Reservation does not necessarily imply that a resource is actually being used. Multiple resources can be reserved with one reservation.

Assignment, like reservation, is process-specific and lasts until unassignment or process termination. Any resource type can be assigned. An assignment also gives a process temporary exclusive rights to a device. Assignment does not necessarily mean that a device is currently being used. That is the function of the next level, attachment. Only one resource can be assigned per assignment.

A resource cannot be used until it is attached. When RCP is called to attach a resource, it initiates communication with the ring 0 subsystem that actually provides the use of the resource. Before the attachment is completed, RCP performs all initialization necessary to allow the attaching process to begin using the resource. For devices, this involves attaching the device via IOI and making sure that the device is ready and that any volume needed has been determined to be accessible, mounted, and authenticated.

The hierarchical relationship among reservation, assignment, and attachment implies that a higher-level function (e.g., reservation) can stand alone, while a lower-level function (e.g., attachment) can only be performed after all higher-level functions have been performed. RCP can perform the following device reservation, assignment, and attachment functions:

1. Reserving a resource. This means that no other process can use it during this period of time.
2. Explicitly assigning a reserved device. The device is assigned to a process but is not attached.
3. Attaching an explicitly assigned device.
4. Attaching an unassigned device. Since a device cannot be attached until it is assigned, RCP automatically reserves and assigns the device and then performs the attachment. The device is said to be implicitly assigned.
5. Detaching an implicitly assigned device. After the device is detached, RCP automatically unassigns the device.
6. Detaching an explicitly assigned device. The device is detached but is not unassigned.
7. Explicitly unassigning a device. If the device is attached, it is first detached and then unassigned.
8. Cancelling reservation of a resource.

The rules stated above imply that I/O modules do not have to be concerned with the assignment or unassignment of devices. They need to be concerned with only the attachment and detachment of a device. RCP, however, does allow the above rules to be overridden. When detaching a device an I/O module can tell RCP to retain the device assignment regardless of whether the device was explicitly or implicitly assigned.

When a process terminates, RCP automatically detaches and unassigns all devices currently assigned to that process and cancels any reservations for that process.



The reservation of resources and cancellation of reservations are done from command level via the `reserve_resource` and `cancel_resource` commands or by using the `-resource` control argument with the `enter_abs_request` command. The explicit assignment and unassignment of devices is done from command level via the `assign_resource` and `unassign_resource` commands. The listing of reservations, assignments, and attachments is done from command level via the `list_resources` command. The `enter_abs_request` command is described in the MPM commands; the other commands named here are described in Section 4 of this manual.

## SUMMARY OF RESOURCE MANAGEMENT FACILITY ACTIONS

Here is a description of the preliminary actions that you may take when you want to use a resource.

### Acquisition

Once a resource is registered by the system administrator, it may be acquired by a user. When a user acquires a resource, he is contracting with the system to become the accounting owner of the resource. In other words, the person who acquires the resource usually agrees to pay a fee for the right to control the access to that resource.

After registering a resource, the system administrator may acquire it in the name of the system or a user, deciding who is allowed to use that resource. Devices (such as tape drives and printers) and "scratch" volumes (e.g., tapes in the system pool) are usually acquired to the system. System-owned resources such as devices and scratch volumes are for use by all users. For other resources such as tape reels and disk packs, the system administrator normally chooses to leave most of these in an unacquired state so that users may acquire these resources on an individual basis.

In order to control the operation of the resource management facility, an administrative table exists that can be adapted to the specific needs of a particular Multics site. This table is referred to as the resource type description table (RTDT). The table is generated from a source language description, called the resource type master file (RTMF), ordinarily prepared by a system administrator. The contents of the RTDT can be examined via the `display_rtdt` command. (The RTDT and the `display_rtdt` command are described in detail in the MAM RCP.)

### Resource Reservation

Users may reserve resources by scheduling with RCP to obtain exclusive rights to a resource for a period of time. RCP enables users to reserve resources or groups of resources through the use of the `reserve_resource` command (described in Section 4). A reservation takes effect immediately and it lasts until either the user's process is terminated or the reservation is specifically cancelled via the `cancel_resource` command (described in Section 4). After invoking `reserve_resource`, the user has exclusive rights to the resource(s).

Tape volumes, tape drives, disk volumes, and disk drives can be reserved. Tape and disk volumes are specified at the time of reservation by name; tape and disk drives are specified by either name or attributes. In the case of disk drives, the only acceptable attribute is `model`. For tape drives, acceptable attributes are `model`, `track`, and `density`. Suitable values for the above-mentioned attributes may be found by using the `list_resource_types` command (described in Section 4).

To cancel reservations, users invoke the `list_resources` command to obtain the reservation identifier, and then invoke the `cancel_resource` command with the reservation identifier to effect the cancellation. Administrators can perform privileged cancellations; that is, if the administrator has proper access, it is possible to cancel reservations belonging to other users.

## Device Assignment

The RCP interface for device assignment allows the caller to request the assignment of a specific device, or any appropriate device of a specified type. To request the assignment of a specific device the caller must ask for the device by name. To request the assignment of an appropriate device of a specified type, the caller must specify the characteristics that the assigned device must have. RCP selects a device for assignment based on the following functional algorithm.

1. If the caller has requested a device by name and if this device is already assigned to the calling process, the assignment is aborted.
2. RCP tests all of the devices of the specified type. RCP counts the number of these devices that are appropriate; appropriate and accessible; and appropriate, accessible and available. These requirements are discussed below:
  - a. appropriate: A device is considered to be appropriate if it has the device characteristics specified by the caller. In testing each device, RCP does not try to match any device characteristics that are not specified by the caller. If a device is asked for by name, only the device name is considered.
  - b. accessible: A device is considered to be accessible if the calling process has rw RCP effective access to the device.
  - c. available: A device is considered to be available for assignment if it is not currently assigned to any process or reserved by another process.
3. Having tested each of these requirements, RCP then makes additional tests to see if a device can be assigned. If the assignment cannot be made, RCP returns an error table code that tells the caller why the assignment aborted. The tests that RCP makes at this time are described below:
  - a. If there are no appropriate devices, the caller is told that the requested resource (device) is not known to RCP.
  - b. If there are no appropriate and accessible devices, the caller is told that he does not have access to the requested resource (device).
  - c. If there are no appropriate, accessible and available devices, the caller is told that the requested resource (device) is not available at this time.
  - d. If this assignment causes the device limits (see "Device Limits" below) to be exceeded, the user is told that he has exceeded the limit.
4. If all the tests described above are passed successfully, the device assignment is made. RCP selects the most advantageous device from the list of devices that were found to be appropriate and accessible and available. It makes this selection based on the following rules:
  - a. If this is a type of device that has volumes and if the caller specified a volume name to use in the device selection and if any device in the list currently has that volume mounted, RCP selects that device.

- b. If the first case is not true, RCP selects the device that has been idle for the longest amount of time.

Having assigned the device, RCP returns all of the characteristics of this device to the caller.

### Device Attachment

The RCP interface for device attachment allows the caller to request a device in the same manner described for device assignment. It can ask for a specific device by name or it can ask for any appropriate device of a specified type. One difference is that if this device is a type that uses volumes, the caller must specify the name of the volume to attach. For assignments, the specification of a volume is optional.

RCP tests all of the devices of the specified type that are already assigned by the requesting process. If the specific device or any appropriate device is already assigned to this process, RCP attaches that device. If no suitable device is already assigned to the requesting process, RCP automatically attempts to assign a suitable device to this process. If no device can be assigned then the attachment is aborted. If the attachment is for a device type that uses volumes, RCP checks to see if the specified volume is already attached to this process or any other process. If the volume is already attached, RCP aborts the attachment.

Once RCP has found a suitable assigned device or has assigned one, it begins the real work of attaching the device. This involves calling IOI to perform the ring 0 device attachment. If the device is a type that uses volumes, RCP tells the operator to mount the specified volume if it is not already mounted on the proper device. Before the attachment is completed, RCP makes sure that the proper volume has been mounted and that any write protection mechanism provided by the device is set correctly. When all of this initialization work has been completed, RCP calls IOI to set the workspace and time-out limits and to promote the validation level of the device. Until this is done, the IOI validation level for the device is the RCP validation level (ring 1). Thus no program in a higher ring can successfully call IOI to use this device until RCP tells IOI to promote it. RCP returns all of the device characteristics of the attached device and all of the information needed to communicate with IOI about this device.

### RESOURCE NAMING CONVENTIONS

While the Multics Resource Management Facility implementation allows resource names to be any ASCII string of up to 32 characters, there are restrictions placed on some of these names by other sources. The remainder of this section describes the details of these resource naming conventions.

#### Device Names

Each device has a unique name. Device names are of one of the following forms:

```
ssss_xx  
dddd
```

The first form is used for devices that share multiplexed I/O channels such as disk and tape devices. The latter is used for all other devices.

In the case of disk and tape device names the name is composed of the subsystem name, 'ssss' in the text above, and the device number, 'xx' above. The subsystem name is defined by the site in the configuration via a PRPH card (see MOH) and the device number is assigned by the Field Engineering Representative when the hardware is installed.

All other devices are also defined in the configuration deck. In this case the PRPH card defines the device itself. These device types include: consoles, printers, card readers, card punches, and special devices.

The four character restrictions listed above are due to the fact that character fields on configuration cards are limited to four characters (one word).

### Volume Names

Volume names are unique within their volume type (e.g., no two tape volumes may have the same name). They may be up to 32 characters in length. The only reserved volume names are "scratch" and "T&D Volume" which are used to designate scratch volumes for disk and tape. A scratch tape is one of the unmarked tapes in an unreserved pool that is used for "scratch"--that is, no information is saved on it from session to session. After every use, it is demounted and returned to the system pool. "T&D Volume" is used for special label processing for online Test and Diagnostics, and its use for attachments requires special privilege.

### ACCESS CONTROL

There are three types of access control on Multics: discretionary access control, which is regulated by access control lists (ACL); nondiscretionary access control, which is regulated by the access isolation mechanism (AIM); and intraprocess access control, which is regulated by the ring structure. (For detailed information on types of access, see the MPM Reference Guide.)

### Access Control Segments

An important feature of RCP is its ability to control access to the various resources that it manages. It does this through the use of access control segments (ACSs). An ACS is a zero length segment whose ACL and ring brackets are used to define the discretionary access to a resource. RCP uses an ACS for each resource that it controls; however, an ACS can be shared by more than one resource. The name of an ACS consists of a name plus the suffix, acs (e.g., tape\_01.acs). There are no restrictions on ACS names other than the required suffix. The user creates an ACS and generates/manipulates its ACL with the create, set\_acl, and delete\_acl commands and ring brackets with the set\_ring\_brackets command (see the MPM Commands).

The pathname of the ACS for a resource is usually specified when it is acquired (see the register\_resource command and the acquire\_resource command in Section 4 of this manual). The specified ACS can later be changed or unspecified so that the resource (again) has no ACS via the set\_resource command (see Section 4). If the ACS has not been specified or does not exist, access is by default rew for the owner of the resource and null for all other users (see access modes in the glossary below).

RCP uses the ACS along with other nondiscretionary controls (AIM) to determine the RCP effective access to a resource.

## Access Class Ranges

Access class ranges are used by RCP to specify that a process within a range of authorizations can use a particular resource.

An access class range is simply a pair of AIM access classes separated by a colon. The first value of the pair is the minimum access class and the second is the maximum access class. If only a single access class is specified when an access class range is expected, the minimum and maximum access class values are both the same (i.e., a range of one value). The second access class of the pair (the maximum) must be greater than or equal to the first (the minimum) according to the `aim_check_` subroutine (see the MPM Subroutines).

There are some interesting results which occur when categories are used in an access class range. For example, a process with authorization of:

```
level2,category1
```

would not be able to use a resource whose access class range was:

```
level1,category1,category2:level3,category1,category2,category3
```

where `level3` is greater than `level2`, which is greater than `level1`. This is due to the fact that the authorization of the process is isolated from the minimum of the access class range. In order to allow this process access to the resource in question, the range would have to exclude `category2` or the user would have to have `category2` authorization. In general, to include categories within an access class range, both the minimum and maximum must include the categories desired. If combinations of categories are desired, the minimum should list only required categories and the maximum should include all categories allowed. For example, the access class range:

```
level1,category1:level3,category1,category2,category3
```

allows read and write access to any `level1`, `level2`, or `level3` process with `category1` and any combination of `category2` and `category3`.

## RCP Effective Access

Viewed separately, each type of access control answers the same question, "What access does a particular process have for a particular item?" The access mode granted a process to a resource by discretionary access control (the ACL) is known as the raw access mode.

The way RCP determines effective access to a resource for a process differs from the regular Multics method of determining effective access as follows. First, the effective access to the ACS for the resource is determined as for any segment. If the ACS does not exist, the user appears to have read, execute, and write access if he is the owner of the resource, or null access if he is not the owner. Then, two further checks are made. First, the current authorization of the process is compared to the maximum access class of the resource. If write access is not allowed (as defined by the `write_allowed_` subroutine) then write and execute access are denied and only read is allowed. Next, the current authorization of the process is compared to the minimum access class of the resource. If read access is not allowed (as defined by the `read_allowed_` subroutine) then all access is denied. The resulting access is termed the RCP effective access to the resource. One final restriction enforced by RCP is that, in order to use a device, the RCP effective access must include both read and write to that device (a restriction not imposed on volumes).

For example, the following table illustrates some examples of RCP effective access. In the examples below, L1, L2, L3 and L4 represent sensitivity levels and c1, c2, c3, and c4 represent categories. (This discussion mostly concerns devices--volumes should never be given multiclassed access class range.)

Table 1-1. RCP Effective Access

Effective Access to ACS	Current Process Authorization	Resource Access Class Range	RCP Effective Access
rew	L1	L1:L3	rew
re	L1	L1:L3	re
rew	L1	L2:L3	null
rew	L3	L2:L3	rew
rw	L4	L2:L3	r
re	L4	L2:L3	r
rw	L2,c1	L1:L4	r
rw	L2,c2	L1,c1:L4,c1,c2	null
rw	L2,c1,c3	L1,c1:L4,c1,c2	r
rw	L2,c1	L1,c1:L4,c1,c2	rw

A user must have write RCP effective access to the resource named to perform any modification on the status of the resource. In addition, the user must have execute effective access to the resource named to modify protected attributes. Only the accounting owner may modify the ACS path.

For more information on AIM, access classes, authorizations, and comparisons involving access classes and authorizations, see the MPM Reference Guide. The access class range mentioned above is specified by the `-access_class` control argument, which can be specified in the `register_resource` command (see the MAM RCP), and the `acquire_resource` and `set_resource` commands (described later in this manual).

### Manipulating RCP Effective Access

Since the access control mechanisms described above operate together to determine the RCP effective access of a process, there are actions that the user, as well as an administrator, can perform to control this effective access.

First, the user creates an ACS via the `create` command. Then, the desired ACL for that segment is established using the `set_acl` command to add desired ACL entries, and the `delete_acl` command to delete entries. (The above three commands are described in the MPM Commands.) To further affect the ACS, the user may modify its ring brackets by using the `set_ring_brackets` command (described in the MPM Subsystem Writers' Guide). The system security administrator sets the AIM access class range of the resource itself at the time it is registered using the `register_resource` command and can change it by using the `set_resource` command.

## ACQUIRING RESOURCES

When a system administrator registers a resource, he may simultaneously cause it to be acquired; that is, designate who will become the accounting owner of that resource. He may immediately acquire it on behalf of either a specific user or to the system. Otherwise, he may leave it in the free pool, and the resource can then be acquired from this pool.

### Acquisition to the System Pool

Resources that are to be owned by the system, such as devices and scratch volumes, may be acquired to the system pool upon registration. System-owned resources are usually set up so that they are shareable among all users. I/O devices, such as tape drives, printers, and punches, are examples of system-owned resources.

### Acquisition by User

Any resource that is not resident in the system or free pools is acquired by a User\_id (Person\_id.Project\_id).

### Acquisition Upon Registration

As a convenience, a system administrator may register a resource and acquire it in one step. The MAM RCP describes this process in detail.

## SITES NOT ENABLING RESOURCE MANAGEMENT

If the system administrator has chosen not to enable Resource Management the preceding discussion of access control can be simplified since no resources can be registered.

Nondiscretionary access control is not enforced in this case. There are no ACSs for volumes, and all users are assumed to have both read and write access to any volume. The ACS for a device can be found in >system\_control\_1>r<sub>cp</sub> and is named device\_name.acs (e.g., tape\_01.acs). Only the discretionary and intraprocess access (ACL and ring brackets) is considered in determining access to a device.

## DEVICE LIMITS

In addition to controlling which processes may have access to a device, RCP will enforce a limit to the number of devices of a given type that a single process may have assigned at one time. This limit is enforced according to the following rules:

1. The limit is not enforced for system processes.
2. The limit for each device type is an installation-defined value. It is currently specified on PRPH (peripheral) configuration cards.
3. Currently, only tape drive devices actually have such a limit defined.

RCP will also enforce a limit to the total number of devices of a given type that may be assigned to non-system processes at one time. RCP enforces this limit in order to ensure that a certain number of devices of each device type are either assigned by a system process or available for assignment by a system process. This limit is enforced according to the following rules:

1. The number of devices of each device type that RCP will reserve for system processes are installation defined values. They are currently specified on PRPH configuration cards.
2. Currently, the only class reserved for system processes is tape drives. Only tape drives with certain characteristics (nine-track tape drives) are reserved, since the backup facility uses only nine-track tapes.

### I/O Workspaces

Due to the nature of the Multics virtual memory and its supporting I/O hardware, I/O operations such as "read tape" or "write disk" require all pages of memory referenced by the I/O operation to be in main memory during the operation--that is, no paging is done during execution of the I/O operation. To accomplish this, all channel programs and physical record buffer areas are located in a special segment known as an I/O workspace segment. The ring 0 I/O software, IOI, guarantees that all pages of the workspace are present in main memory before starting the I/O operation and remain there for the duration of the operation.

RCP will control the maximum workspace size associated with each device type. System processes, privileged processes, and users on the ACL of the ACS named workspace.acs in the directory >system\_control\_1>rcp can request up to the privileged maximum workspace size. All others can request up to the normal maximum workspace size. Requests for a workspace larger than is allowed result in errors. The table below lists the workspace maximums that are enforced.

Table 1-2. I/O Workspaces

Privileged Maximum			Normal Maximum	
device type	words	bytes	words	bytes
tape_drive	45056	180224	3072	12288
disk_drive	45056	180224	2048	8192!
printer	45056	180224	1024	4096!
punch	45056	180224	1024	4096!
reader	45056	180224	1024	4096!
special	45056	180224	1024	4096!
console	45056	180224	1024	4096!

The workspace size is affected by using the -block control argument to those I/O modules that support it. This control argument is used to specify the maximum physical record/block size to be processed. In all cases some overhead for channel programs and I/O module control information must be taken into consideration. When -block is not specified or supported the individual I/O modules choose an appropriate default. In the case of commands that use I/O modules, either the command, some argument or input to the command, or the I/O module may specify/imply in some way the workspace size (for example by supplying -block in an attach description).



## GLOSSARY

### access class

An access isolation mechanism (AIM) attribute that denotes the sensitivity of information contained in a segment, directory, multisegment file; message in a message segment; contained on a volume; or processed using a device. An access class is associated with an entry for its lifetime.

### access class range

The access range defined by two access classes, including the access classes between them. See "Access Class Ranges" described above.

### access control

The method for determining who may reference or modify segments (files) and directories, or use a resource.

### access control list (ACL)

A set of access identifiers specifying who may access a segment or directory. Associated with each access identifier is a set of allowed modes of access to that segment or directory. There is an ACL for each segment and each directory.

### access control segment (ACS)

A zero length segment whose ACL is used in determining the user's access to a resource (see "Access Control Segments" described above).

### access isolation mechanism (AIM)

Extension of standard access controls. AIM regulates the nondiscretionary access control features, which allow site-defined levels and categories of privilege.

### access modes

A way to identify the kinds of access that may be set for a resource. The access modes for segments are read (r), write (w), execute (e), and null (n). Those for directories are status (s), modify (m), append (a), and null (n).

### accounting owner

The person who usually pays a fee for the right to control the access to a resource for a certain period of time (also referred to as owner).

### acquire

To contract with the system to become the accounting owner of a resource. Acquisition is not per process; it lasts until the resource is explicitly released (see release).

### allocation switch

A switch, strictly for the user's convenience, that is turned on or off. For example, if the resource is a tape volume, it can be used to indicate whether the tape contains needed data.

### assign

To allow a process to have temporary exclusive rights to a resource (no one else can use it during this period of time -- no longer than per-process).

### attributes

Fields that give descriptive information about devices and volumes. For example, a tape can have the attributes track=7 and den=1600, which describe the density and number of tracks on a tape. Allowable values for attributes are defined by the system administrator. Attributes can be listed by users with the resource\_status command and changed by users with the set\_resource command (see Section 4). (See also potential attributes and protected attributes below.)

cancel To give up exclusive rights to a resource; the opposite of reserve.

deregister To remove a resource from the free pool; makes it unknown to the system (see register).

free pool The list of resources kept by resource management from which users may acquire resources.

free resource A resource that is in the free pool.

known In RCP, to register a resource is to make it known to the system (at that time, the resource is entered into one of the registries).

potential attributes Fields that show what the resource is capable of doing. For example, a tape drive can have the potential attribute track=7 or track=9 but not both. It might also have the potential attributes den=1600 and den=6250 but not den=800.

protected attributes Fields that designate that certain attributes of a device are protected from being changed. The user must have execute effective access to the resource to modify protected attributes.

RCP administrator An administrator who can register and deregister resources. To do this he must have execute access to the rcp\_admin\_gate.

RCP effective access A user's access to a resource, determined by a combination of access-regulating methods (see "RCP Effective Access" in this section).

register To introduce a resource to the system as available for acquisition; a registered resource remains so until explicit deregistration--this is not a per-process state.

registry A data base maintained by and for RCP that describes all resources of a given type. There is a registry for each type of resource.

release To give up ownership of a resource and put it back into the free pool (opposite of acquire).

reserve To schedule with RCP the exclusive rights to a resource.

resource A component of the system, such as a tape drive or a volume used on one of those peripherals whose use is controlled by RCP.

resource control package (RCP) The software, operating in ring 1, that controls the registration, acquisition, reservation, assignment, and use of the resources whose resource types are described in the RTDT.

resource type A category of resources, such as tape\_drive or printer.

resource type description table (RTDT) A binary segment describing in detail all of the resource types (e.g., tape\_drive, tape\_vol) controlled by RCP.

resource type master file (RTMF)  
An ASCII segment compiled by the cv\_rtmf command to produce the RTDT.

ring  
A particular level of privilege at which programs may execute. Lower-numbered rings are of higher privilege than higher-numbered ones. The supervisor program runs in ring 0; most user programs runs in ring 4.

ring brackets  
A set of integers associated with each segment that define in what rings that segment may be written, read, called, or executed.

system administrator  
A highly privileged user who maintains system data bases that control when and by whom the system can be accessed. The system administrator has access to all Multics commands, has the ability to alter any operating parameter of the system, and may make emergency repairs. He is also concerned with the basic rules and prices for use of system resources.

system pool  
The list of system-owned resources (such as tape drives, scratch volumes, and disk drives) that may be used on a public basis.

system security administrator  
A system administrator whose primary responsibility is the integrity of the system and maintenance of the access control mechanisms, particularly AIM.

unassign  
To give up temporary exclusive rights to an (unreserved) resource; opposite of assign.

User\_id  
A Person\_id.Project\_id pair designating ownership of a resource, for accounting purposes.

## SECTION 2

### HOW TO USE RESOURCE MANAGEMENT

As an aid to understanding how users interact with the Resource Management Facility, this section shows by example how a user at a sample site requests and manipulates tapes. Although the examples refer specifically to tapes, the methods generally apply to the use of other resources as well.

View this section as an imaginary publication issued at a specific site. Certain capabilities, which have been marked nonstandard, have been developed by administrators at the site and are therefore site-specific.

#### USER'S GUIDE TO TAPES AT SAMPLE SITE

##### Tape Registration

Before a tape may be used, it must be registered. An attempt to mount a tape that has not been registered results in an error message such as "Resource not known to system" or "Mount request not honored." To verify that a tape is registered, use the `resource_status (rst)` command. For example, to see if tape EX0010 is registered, use the following command line:

```
resource_status tape_vol EX0010
```

##### Acquisition Of Tapes

To request a tape to use (i.e., to rent) use the `acquire_resource (aqr)` command:

```
acquire_resource tape_vol -number 1
```

The system returns the name of a tape which is then "owned" by the user.

To release a tape, use the `release_resource (rlr)` command:

```
release_resource tape_vol reel_no
```

To list all tapes that have been acquired (i.e., for which you are the owner) use the `list_resources (lr)` command:

```
list_resources -acquisitions -type tape_vol
```

To use a tape from another computer system, you must have it registered. (The following describes a sample site operating policy.) Give the tape to the Multics operator. If you know what label is recorded on the tape, give that information to the operator. If you do not know what the label is, the operator assigns a number for the tape. You must also tell the operator your Multics Person\_id and Project\_id. The operator then registers the tape and acquires it to you. You receive a message when this has been done. (The sending of this message is a nonstandard, site-specific feature.) You can also verify that this has been done by listing your tapes with the list\_resources command as explained above. You may then use the tape exactly as you would a tape that you had acquired with acquire\_resource, except that you cannot release the tape. (The fact that you cannot release the tape is also a nonstandard, site-specific item.) The same warnings apply about access. By default, only the owner can use the tape. When you have finished using the tape, ask the operator to deregister the tape (thereby removing you as the owner) and to return the tape to you.

### Access Control

Note that by default, the access for a tape is "rw Person.Project", where Person and Project are the Person\_id and Project\_id of the owner of the tape. This means that this tape is only usable by that person, and only under that project. An attempt to use the tape under a different Person\_id or Project\_id results in an error message stating "Resource not accessible to requesting process" or "Mount request not honored." Check your access to a tape by using the resource\_status (rst) command as described above.

To change this access, the owner of a tape can use the set\_resource command. With set\_resource, specify an Access Control Segment (ACS), whose ACL is used as the ACL of the tape. This ACS is not created by the set\_resource command. It must be created by the user.

```
create path.acs
set_acl path.acs mode User_id
set_resource tape_vol tape -acs_path path
```

For example, to give user Jones access to write on a tape named CU0001 and allow anyone on the XYZ project to read the tape, the owner of tape CU0001 could use the following sequence of commands:

```
create my_tapes.acs
set_acl my_tapes.acs rw Jones.*
set_acl my_tapes.acs r *.XYZ
set_resource tape_vol CU0001 -acs_path my_tapes.acs
```

## SECTION 3

### ADMINISTRATIVE AND USER INTERFACES

#### ADMINISTRATIVE DATA BASES AND INTERFACES

Several data bases and administrative commands are required to manage resources via the RCP facility. If resource management is not activated, these features can be ignored. However, once resource management is enabled, the RCP administrator must manage the data bases and perform privileged actions for the user community.

The sequence of events that must occur to use a resource with resource management enabled is:

1. The RCP administrator registers the resource using the `register_resource` command, making the resource known to the system.
2. The user acquires the resource using the `acquire_resource` command, telling the system to make him the owner and stating his willingness to pay for the resource (this can also be done by the administrator for the user).
3. Now the resource may be used by any user with appropriate access.

A variety of information is stored by the system as part of resource management. This information is under the control of the RCP administrator. This includes all of the information in the resource type description table (RTDT) and most information in the registries. The RTDT, which is generated by the RCP administrator, defines all of the resource types known by the system. Also defined in the RTDT are default values for the potential attributes, the potential access class range, and the charge type to be used in billing for resources of a given type. The registries contain information specified by the administrator at registration time or when a resource is acquired for a user. The RTDT and the registries are described in detail in the MAM RCP.

#### USER INTERFACES

##### Acquiring and Using Resources

Once the RCP administrator has registered a resource, a user may acquire it (if it is not already acquired), or, the administrator can do this acquisition for a user. Acquisition tells the system that the user is willing to pay for the resource(s) specified--for example, a tape volume. For more information on acquisition, see the `acquire_resource` command in Section 4.

Once a resource has been acquired it can be used (reserved, assigned, and attached) by any user with appropriate access. This means the user is on the ACL of the ACS, is at the proper validation level, and is within the access class range for this resource specified by the administrator. (See "Access Control Segments" in Section 1.)

It is important to realize that there is normally no implicit acquisition, and that only acquired resources can be used. The only exception to these rules occurs when a site has "automatic registration" turned on during the initial time period after enabling the full resource management facility. While automatic registration is on, any unregistered tape volume for which the operator honors a mount request is automatically registered and acquired to the requesting user.

### Naming Rules for Attributes

Attributes provide a description of a volume or device that assists the resource management facility in the proper matching of volumes with compatible devices. To produce correct combinations, attribute names must comply with the set of rules described below.

Attributes may be grouped or ungrouped. Grouped attributes specify a set of properties applicable to a device or volume such that only one attribute of that set can be currently active at any given time. For example, a reel of tape may have potential attributes that allow it to be recorded at densities of 556, 800, or 1600; however, at any given time, the data on it is in only one of those densities. Grouped attributes have names of the form:

<identifier>=<value>

For example, the attributes mentioned above are named "den=556", "den=800", and "den=1600". This notation allows RCP to recognize that any request to make one of these attributes the current attribute of a device or volume also implies that all other attributes in that grouping must be made inactive.

Ungrouped attributes have simple names, such as "trainok" (to specify that this device accepts a removable print train) or "building\_12" (to specify that this device or volume is located in building 12).

### User Manipulation of Registries

The user of a resource has limited control over the information maintained by the system describing the resource. Most of this information is specified/modified via the `acquire_resource` and `set_resource` commands. When a user acquires a resource (using the `acquire_resource` command), he can specify the ACS pathname, current attributes, comment, and user allocation state of the resource using the `-acs_path`, `-attributes`, `-comment`, and `-alloc` control arguments. At any time between acquisition and release, this information can be manipulated using the `set_resource` command and displayed using the `resource_status` command/active function. See Section 4 for descriptions of these commands.

Any user with write RCP effective access to the resource can modify the current attributes, comment, or user allocation state. In addition, only resource executives (execute RCP effective access to the resource) can modify protected attributes. Finally, only the owner of the resource can modify the ACS pathname maintained by resource management.

However, RCP administrators, by virtue of their privilege, can effectively modify all the information described, for any resource, plus other information not described here (see the MAM RCP for details).



## SECTION 4

### USER COMMANDS

This section contains descriptions of the user commands that acquire resources, list all resource types described in the RTDT, release resources, print a description of the status of a resource, and modify the parameters of a resource. Each description contains the name of the command (including the abbreviated form, if any), discusses the purpose of the command, and shows the correct usage. Notes and examples are included when deemed necessary for clarity. The discussion below briefly describes the content of the various divisions of the command descriptions.

#### Name

The "Name" heading lists the full command name and its abbreviated form. The name is usually followed by a discussion of the purpose and function of the command and the expected results from the invocation.

#### Usage

This part of the command description first shows a single line that demonstrates the proper format to use when invoking the command and then explains each element in the line. The following conventions apply in the usage line.

1. Optional arguments are enclosed in braces (e.g., {path}, {User\_ids}). All other arguments are required.
2. Control arguments are identified in the usage line with a leading hyphen (e.g., {-control\_args}) simply as a reminder that all control arguments must be preceded by a hyphen in the actual invocation of the command.
3. To indicate that a command accepts more than one of a specific argument, an "s" is added to the argument name (e.g., paths, {paths}, {-control\_args}).

NOTE: Keep in mind the difference between a plural argument name that is enclosed in braces (i.e., optional) and one that is not (i.e., required). If the plural argument is enclosed in braces, clearly no argument of that type need be given. However, if there are no braces, at least one argument of that type must be given. Thus "paths" in a usage line could also be written as:

```
path1 {path2 ... pathn}
```

The convention of using "paths" rather than the above is merely a method of saving space.

4. Different arguments that must be given in pairs are numbered (e.g., xxx<sub>1</sub> yyy<sub>1</sub> {... xxx<sub>n</sub> yyy<sub>n</sub>}).
5. To indicate that the same generic argument must be given in pairs, the arguments are given letters and numbers (e.g., pathA<sub>1</sub> pathB<sub>1</sub> {... pathA<sub>n</sub> pathB<sub>n</sub>}).
6. To indicate one of a group of the same arguments, an "i" is added to the argument name (e.g., path<sub>i</sub>, User<sub>idi</sub>).

To illustrate these conventions, consider the following usage line:

```
command {paths} {-control_args}
```

The lines below are just a few examples of valid invocations of this command:

```
command
command path path
command path -control_arg
command -control_arg -control_arg
command path path path -control_arg -control_arg -control_arg
```

In many cases, the control arguments take values. For simplicity, common values are indicated as follows:

STR	any character string; individual command descriptions indicate any restrictions (e.g., must be chosen from specified list; must be either the string on or the string off).
N	number; individual command descriptions indicate whether it is octal or decimal and any other restrictions (e.g., cannot be greater than 4).
DT	date-time character string in a form acceptable to the convert_date_to_binary_ subroutine described in the MPM Subroutines.
path	pathname of an entry; unless otherwise indicated, it may be either a relative or an absolute pathname.

The lines below are samples of control arguments that take values:

```
-access_name STR, -an STR
-ring N, -rg N
-date DT, -dt DT
-home_dir path, -hd path
```

### Notes

Comments or clarifications that relate to the command as a whole are given under the "Notes" heading. Also, where applicable, the required access modes, the default condition (invoking the command without any arguments), and any special case information are included.

## Examples

The examples show different valid invocations of the command. An exclamation mark (!) is printed at the beginning of each user-typed line. This is done only to distinguish user-typed lines from system-typed lines. The results of each example command line are either shown or explained.

## Other Headings

Additional headings are used in some descriptions, particularly the more lengthy ones, to introduce specific subject matter. These additional headings may appear in place of, or in addition to, the notes.

Name: acquire\_resource, aqr

The acquire\_resource command selects a resource of a given type from a free pool of all such resources, and makes the user the accounting owner of the resource. The accounting owner is given full control over the access rights of all users of the resource, as well as control over many parameters of the resource. Ownership of the resource is terminated via the release\_resource command.

### Usage

```
acquire_resource type STR1 {... STRn} {-control_args}
or
aqr type -number N {-control_args}
```

where:

1. type is a resource type defined in the RTDT.
2. STR<sub>i</sub> is the unique identifying name of the particular resource being acquired. If STR looks like a control argument (i.e., if it is preceded by a hyphen), then it must be preceded by -name or -nm. If name is not supplied, -number must be given, and a resource is chosen to satisfy the constraints imposed by the control arguments given.
3. control\_args can be chosen from the following:
  - access\_class accr, -acc accr sets the initial AIM access class parameters where accr is an access class range. Users at any authorization within the access class range inclusive are allowed to read and write to the resource (provided they also meet other access requirements). For a detailed description see "Access Class Ranges" in Section 1.
  - acs\_path path specifies the pathname of the access control segment (ACS) for this resource. The ACS is not created by this command, but must be created by the owner, and the desired access control list set. If the ACS does not exist or is not specified, the default access is rew to the accounting owner, and null to all others.
  - alloc STR sets the allocation state of the resource to free or allocated, where STR must be either the string on or the string off. If this control argument is not given, the allocation state is free. (The allocation state flag is a convenience to the user and is largely ignored by resource management.)
    - on sets the allocation state to allocated
    - off sets the allocation state to free

- attributes STR, -attr STR  
specifies that the resource chosen must possess the potential attributes specified in STR. When a satisfactory resource is located, the current attributes are set to a proper combination of these attributes (see "Notes" below).
- comment STR, -com STR  
specifies the desired value of the comment string for this resource.
- lock STR  
locks or unlocks the resource, preventing or allowing use of that resource, where STR must be either the string on or the string off. If this control argument is not specified, the lock is off.  
on prevents any use of the resource  
off allows use of the resource
- number N, -nb N  
specifies that the number of such resources to be acquired is N. If this control argument is not given, 1 is assumed. This control argument may only be specified if the STR argument is not given (see "Usage" above).
- owner STR, -ow STR  
specifies that this is an acquisition on behalf of the user specified by STR. If STR is the string "system", then the resource is acquired to the system pool. If STR is the string "free", then the resource is left in the free pool (effectively the same as no -owner). If STR is of the form Person\_id.Project\_id (where neither Person\_id nor Project\_id may be a star), the user specified has all the rights of ownership to the resource as if he had acquired it personally, except that if "-release\_lock on" is specified, the owner may not release (give up ownership of) the resource voluntarily.
- release\_lock STR, -rll STR  
specifies whether this resource may be released by the owner, or may only be released by a privileged process (see "Access Restrictions" below), where STR must be either the string on or the string off. If this control argument is not specified or is the string off, the resource may be released by the owner.  
on resource may only be released by privileged process  
off resource may be released by owner

---

acquire\_resource

---

---

acquire\_resource

---

### Notes

This command acquires a resource for either the user issuing it (requestor) or the user specified by the `-owner` control argument. If the requestor is registered on more than one project and needs corresponding access, or other users (on any project) need access to use a resource, the owner must create or modify an access control segment (ACS). The owner must then specify the new/modified ACS using the `-acs_path` control argument either when issuing this command or later with the `set_resource` command.

For a description of the syntax of attribute strings, see "Naming Rules for Attributes" Section 3.

The use of this command for attaining the desired RCP effective access is explained under "Access Control Segments" in Section 1.

### Access Restrictions

The use of the `-owner`, `-release_lock`, or `-access_class` control arguments requires execute access to the `rcp_admin_gate`.

Name: `assign_resource`, `ar`

The `assign_resource` command calls the resource control package (RCP) to assign a resource to the user's process.

### Usage

```
ar resource_type {-control_args}
```

where:

1. `resource_type`

specifies the type of resource to be assigned. Currently, only device types can be specified. The `-device` control argument is used to name a specific device to assign. Other control arguments are used to specify characteristics of the device to be assigned. The following device type keywords are supported:

```
tape_drive
disk_drive
console
printer
punch
reader
special
```

2. `control_args`

can be chosen from the following:

`-device STR`, `-dv STR`

specifies the name of the device to be assigned. If this control argument is specified, other control arguments that specify device characteristics are ignored. (See "Examples" below.) If the `-long` control argument (see below) is used in conjunction with this control argument, a message containing the name of the assigned device is printed on the user's terminal; otherwise, no message is printed.

`-model N`

specifies the device model number characteristic. Only a device that has this model number is assigned. In order to find the model numbers that are acceptable, use the `print configuration_deck` command described in System Tools, Order No. AZ03.

`-track N`, `-tk N`

specifies the track characteristic of a tape drive. The value can be either 9 or 7. If this control argument is not specified and if the `-volume` control argument is not specified, a track value of 9 is used when assigning a tape device.

- density N, -den N**  
specifies the density capability characteristic of a tape drive. There can be more than one instance of this argument. A tape drive is assigned that is capable of being set to all of the specified densities. The acceptable values for this argument are:  
200  
556  
800  
1600  
6250
- Note that the values permitted depend on the particular hardware on the system.
- train N, -tn N**  
specifies the print train characteristic of a printer.
- line\_length N, -ll N**  
specifies the line length of a printer. Its value must be one that is found in the "line length" field of a printer PRPH configuration card. If this field is not specified on a printer PRPH configuration card, this device characteristic is ignored for this printer.
- volume STR, -vol STR**  
specifies the name of a volume. If possible, the device assigned is one on which this volume has already been placed. If this is not possible (e.g., the volume is on a device assigned to a process) any available, appropriate, and accessible device will be assigned.
- number N, -nb N**  
specifies the number of resources to assign. All of the resources assigned have the device characteristics specified by any other arguments passed to this command. If this control argument is not specified, one resource is assigned.
- comment STR, -com STR**  
is a comment string that is displayed to the operator when the resource is assigned. If more than one string is required, the entire string must be in quotes. Only printable ASCII characters are allowed. Any unprintable characters (also tabs or new lines) found in this string are converted to blanks.
- long, -lg**  
specifies that all of the device characteristics of the assigned device should be printed. If this argument is not supplied, only the name of the assigned device is printed.
- system, -sys**  
specifies that the user wants to be treated as a system process during this assignment. If this argument is not specified or if the user does not have the appropriate access, then the RCP assumes that this assignment is for a nonsystem process.
- wait {N}, -wt {N}**  
specifies that the user wants to wait if the assignment cannot be made at this time because the resources are assigned to some other process. The value N specifies the maximum number of minutes to wait. If N minutes elapse and a resource is not yet assigned, an error message is printed. If N is not specified, it is assumed that the user wants to wait indefinitely.



-speed N  
specifies the speed of a tape drive. The acceptable values depend on the particular hardware on the system and can be the following:  
75  
125  
200

### Notes

Currently, only device resources can be assigned. An assigned device still must be attached by a call to some I/O module. If a device is successfully assigned, the name of the device is printed. (If the user requests a specific device that is successfully assigned, the name of the device is not printed unless the user asks for it. See the -device and -long control arguments above.)

### Examples

In the example below, the user issues the assign\_resource command with the "tape\_drive" keyword and the -model control argument. The system responds with the name of the assigned device.

```
! assign_resource tape_drive -model 500
```

```
Device tape_04 assigned
```

In the next example, the user issues the assign\_resource command with the "tape\_drive" keyword and the -device and -long control arguments. The system responds with the name of the assigned device and the model number, track, density and speed characteristics.

```
! assign_resource tape_drive -device tape_05 -long
```

```
Device tape_05 assigned  
Model      = 500  
Tracks     = 9  
Densities  = 200 556 800 1600  
Speed      = 125
```

---

cancel\_resource

---

---

cancel\_resource

---

Name: cancel\_resource, cnr

The cancel\_resource command cancels reservations made with the reserve\_resource command using the reservation identifier obtainable from the list\_resources command.

Usage

cnr -id reservation\_id {-control\_arg}

where:

1. reservation\_id  
must be present and is the reservation identifier of the reservation to be cancelled. It must be preceded by the -id control argument.
2. control\_arg  
can be -priv to specify the privileged cancellation of a reservation belonging to another user (see "Access Restrictions" below).

Notes

Reservation identifiers can be obtained by using the list\_resources command.

Access Restrictions

Use of the -priv control argument requires access to rcp\_sys\_.

---

list\_resource\_types

---

---

list\_resource\_types

---

Name: list\_resource\_types, lrt

The list\_resource\_types command prints a list of all resource types described in an RTDT.

Usage

```
list_resource_types {type1 ... typen} {-control_args}
```

where:

1. type<sub>i</sub>  
is the resource type defined in the RTDT for which information is to be listed. If no type is specified, all known resource types are listed.
2. control\_args  
can be chosen from the following:
  - long, -lg  
lists the defined attributes for each resource type.
  - no\_header, -nhe  
omits the column headers.
  - pathname path, -pn path  
lists resource types defined in the RTDT specified by path. If this control argument is not specified, the RTDT residing in >system\_control\_1 is used.

Name: `list_resources, lr`

The `list_resources` command lists groups of resources managed by the Resource Control Package (RCP), selected according to criteria specified by the user.

Usage

`lr {-control_args}`

or

`[lr {-control_args}]`

where `control_args` can be chosen from the following:

- `-acquisitions, -acq`  
lists resources acquired by the user specified by the `-user` control argument. If this control argument is used, `-type` must also be specified.
- `-assignments, -asm`  
lists resource assignments. This cannot be used with the active function.
- `-awaiting_clear`  
lists those resources that are awaiting manual clearing. If this control argument is used, `-type` must also be specified.
- `-device STR, -dv STR`  
lists device resources with the name `STR`. No other resources are listed. This cannot be used with the active function.
- `-logical_volume, -lv`  
lists logical volumes that are currently attached. This cannot be used with the active function.
- `-long, -lg`  
prints all the information known about each resource listed. If this control argument is not supplied, only the name is printed for each resource listed. This cannot be used with the active function. `-lg` has no effect if the `-acq` control argument has been specified.
- `-mounts, -mts`  
lists resources currently mounted by the process. This cannot be used with the active function.
- `-reservations, -resv`  
lists only device and volume reservations. This cannot be used with the active function.
- `-type STR, -tp STR`  
lists resources of the type `STR`. See `list_resource_types` for information on obtaining the names of resource types.

-user User\_id  
selects a particular user or group of users for whom resource information is to be printed. This control argument can be used only in conjunction with -acquisitions. The User\_id can be any of the following forms:

Person.Project  
specifies a particular Person\_id and Project\_id combination.

\*.Project  
specifies all users on a specified project.

\*. \*  
specifies all users (i.e., all acquired resources of the given type are listed).

free  
specifies all resources of the given type in the free pool.

system  
specifies all resources of the given type in the system pool.

\*\*  
specifies all users plus the free and system pools (i.e., all registered resources of the given type will be listed).

If this control argument is not specified, the User\_id of the user invoking list\_resources is assumed. If this control argument is used, -type must also be specified. See "Notes on Access Restrictions" below.

### Notes

If this command is invoked without any arguments, all resources assigned and devices attached to the calling process are listed.

Access to rcp\_admin\_ is required to obtain information on other users' resources. Read access to the PDT (Project Definition Table) of a specified project is required to obtain information for that project.

### Examples

In the example below, the user issues the list\_resources command with no control arguments. The system responds with the name of the assigned devices.

```
! lr
Device Assignments
Device tape_05
Device tape_02
```

In the next example, the user issues the list\_resources command with the -long control argument. The system responds with all the information known about each resource listed.

```
! lr -lg
```

```
Device Assignments
```

```
2 devices assigned
```

```
Device tape_05
```

```
State = assigned  
Time = 04/30/76 1316.2 edt Fri  
Disp = retain  
Level = 4  
Model = 500  
Tracks = 9  
Densities = 200 556 800 1600  
Speed = 125
```

```
Device tape_02
```

```
State = assigned  
Time = 04/30/76 1314.7 edt Fri  
Disp = retain  
Level = 4  
Model = 500  
Tracks = 9  
Densities = 200 556 800 1600  
Speed = 125
```

In the following example list\_resources is invoked to obtain a list of tape volumes for which the user is the accounting owner.

```
! lr -tp tape_vol -acq
```

```
4 resources of type tape_vol acquired by Dahl.GNP  
at 05/10/79 2025.5 mst Thu:
```

```
a-153  
a-022  
u-405  
a-558
```

---

release\_resource

---

---

release\_resource

---

Name: release\_resource, rlr

The release\_resource command releases a resource into the free pool. A resource may only be released by its accounting owner or an RCP administrator.

Usage

release\_resource type STR<sub>1</sub> ... STR<sub>n</sub> {-control\_arg}

where:

1. type is a resource type defined in the RTDT.
2. STR<sub>i</sub> is the unique identifying name of the particular resource being released. If STR looks like a control argument (i.e., if it is preceded by a hyphen), then it must be preceded by -name or -nm.
3. control\_arg can be -priv to specify that the user wishes to perform a privileged release of this resource from the accounting owner, even though the user may not be the accounting owner (see "Access Restrictions" below).

Access Restrictions

The use of the -priv control argument requires execute access to the rcp\_admin\_gate.

---

reserve\_resource

---

---

reserve\_resource

---

Name: reserve\_resource, rsr

The reserve\_resource command reserves a resource or group of resources for use by the calling process. The reservation takes effect immediately and lasts until cancelled by the cancel\_resource command or by process termination. The user pays for reserved time as if the resource were attached.

### Usage

rsr -control\_arg

where control\_arg is -resource STR, -rsc STR, to specify a description of the resources to be reserved. If this resource description contains spaces or special characters, it must be enclosed in quotes. This resource description can also have control arguments and is described in more complete detail below.

### Notes on Resource Description

A resource description describes certain devices and volumes by name or by attributes and an optional number. It has the following format:

```
{-resource_type} resource_spec1 ... {-resource_type resource_specn}
```

That is, a series of at least one resource\_spec where all but the first must be preceded by the -resource\_type or -rsct control argument. The first need not be preceded by the control argument.

The format of a resource\_spec is as follows:

```
volume_type name1 {names}
```

or:

```
device_type {names}
```

or:

```
device_type {-control_args}
```

where:

1. volume\_type  
can be either tape\_vol or disk\_vol. At least one name must be specified with volume\_type, and it is the name of the volume, for example, 050102.
2. device\_type  
can be either tape\_drive or disk\_drive. Names are the names of devices such as tape\_01, and if names are specified with a device\_type, no control arguments are allowed. Names are not required, therefore a device\_type can be specified with no names or control arguments. If names are not specified, the control\_args for use with device\_type can be chosen from the following:



-attributes STR, -attr STR  
the attribute string STR consists of a string of attributes with values separated by commas with no spaces. For tape drives the attributes are:

mode=  
track=  
den=

For disk drives the only attribute is:

model=

Suitable values for these attributes can be found by using the list\_resource\_types command also in this manual.

-number N, -nb N  
is the number of identical resources of the type desired.

### Examples

```
rsr -rsc "tape_vol 50102 u-309 -rsct tape_drive -attr  
track=9,den=800 -nb 2"
```

This command line reserves four resources: two tapes, 050102 and u-309; two tape drives, both being 9-track and capable of 800 bpi operation.

Name: resource\_status, rst

The resource\_status command prints selected information about the status of a given resource. This command can also be invoked as an active function (see "Notes" below).

### Usage

```
resource_status type STR1 ... STRn {-control_args}
or
[resource_status type STR -control_arg]
```

where:

1. type  
is a resource type defined in the RTDT.
2. STR<sub>i</sub>  
is the unique identifying name of the particular resource desired. If STR looks like a control argument (i.e., if it is preceded by a hyphen), then it must be preceded by -name or -nm.
3. control\_args  
can be chosen from the following:
  - access\_class, -acc  
prints the AIM access class or access class range of the resource (see "Notes" below).
  - acs\_path  
prints the pathname of the ACS for this resource (see "Notes" below).
  - all, -a  
specifies that all information maintained about this resource is to be printed. This control argument is not allowed in an active function invocation.
  - alloc  
specifies that the state of the user allocation switch for this resource is to be printed.
  - attributes, -attr  
prints the current and protected attributes of this resource.
  - comment, -com  
prints the user-settable comment associated with this resource.
  - location, -loc  
prints the location field associated with this resource.
  - lock  
prints the status of the resource lock for this resource. In an active function invocation, "true" is returned if the lock is on; "false" is returned if it is off.

- mode, -md  
prints the user's RCP effective access mode to the resource.
- owner, -ow  
prints the name of the owner of the resource.
- potential\_access\_class, -pacc  
prints the potential access class or potential access class range for this resource (see "Notes" below).
- potential\_attributes, -pattr  
prints the potential attributes of this resource.
- priv  
specifies that a privileged call is to be made to obtain the status of this resource (see "Access Restrictions" below).
- release\_lock, -rll  
prints the status of the lock which prevents the owner from releasing this resource. In an active function invocation, "true" is returned if the lock prevents the owner from releasing the resource; "false" is returned otherwise.
- uid  
prints the unique identifier of this resource.

### Notes

If no control arguments are given, some fields are printed by default. The information about the specified resource that this command prints by default is the name of the owner, the current and protected attributes, the user's RCP effective access mode, the pathname of the ACS, the user\_settable comment, and the state of the user allocation switch for the specified resource.

When invoked as an active function, this command returns the value requested by the specified control argument (only one control argument may be specified in this usage). (Active functions are fully described in Section 2 of the MPM Commands.)

### Access Restrictions

The use of the -priv control argument requires execute access to the rcp\_admin\_gate.

Name: set\_resource, setr

The set\_resource command is used to modify parameters of a resource.

Usage

set\_resource type STR<sub>1</sub> ... STR<sub>n</sub> -control\_args{s}

where:

1. type is a resource type defined in the RTDT.
2. STR<sub>i</sub> is the unique identifying name of the particular resource being modified. If STR looks like a control argument (i.e., if it is preceded by a hyphen), then it must be preceded by -name or -nm.
3. control\_args can be chosen from and must include at least one of the following:
  - access\_class accr, -acc accr  
sets the initial AIM access class parameters, where accr is the access class range. Users at any authorization within the access class range inclusive are allowed to read and write to the resource (provided they also meet other access requirements). For a detailed description see "Access Class Ranges" in Section 1.
  - acs\_path path  
specifies the pathname of the access control segment (ACS) for this resource. The ACS is not created by this command, but must be created by the accounting owner, and the desired access control list set (see "Notes" below). If this control argument is not given, the accounting owner of the resource is given rew access by default. If path is a null string, the existing ACS, if any, is disassociated from the resource.
  - alloc STR  
sets the allocation state of the resource to free or allocated, where STR must be either the string on or the string off. If this control argument is not given, the allocation state is free. (The allocation state flag is a convenience to the user and is largely ignored by resource management.)  
on sets the allocation state to allocated  
off sets the allocation state to free
  - attributes STR, -attr STR  
specifies the desired values for the attributes of this resource (see "Notes" below).
  - comment STR, -com STR  
specifies the desired value of the comment string for this resource.
  - location STR, -loc STR  
specifies a descriptive location for the resource, to aid the operator in locating it when it is stored in a special place (e.g., a vault, a different room, etc.).

- lock STR**  
locks or unlocks the resource, preventing or allowing use of that resource, where STR must be either the string on or the string off. If this control argument is not specified, the lock is off.  
on prevents any use of the resource  
off allows use of the resource
- release\_lock STR, -rll STR**  
specifies whether this resource may be released by the owner, or may only be released by a privileged process (see "Access Restrictions" below). If this control argument is not specified or is off, the resource may be released by the owner (does not require special privilege).  
on resource may only be released by privileged process  
off resource may be released by owner

### Notes

If multiple resources are specified to the set\_resource command and an error occurs in the modification of one of these resources, none of the resources specified are modified.

For a description of the syntax of attribute strings, see "Naming Rules for Attributes" in Section 3.

### Access Restrictions

The user must have write RCP effective access to the resource named to perform any modification on the status of the resource. In addition, the user must have execute effective access to the resource named to modify protected attributes (described in Section 1). Only the accounting owner may modify the ACS path. The user must have execute access to the rcp\_admin\_gate in order to use the -access\_class, -release\_lock, -location, -charge\_type, or -lock control arguments.

Certain specifications of AIM access class parameters (e.g., an access class lower than the user's current authorization) are rejected unless the user has the AIM rcp privilege.

Name: unassign\_resource, ur

The unassign\_resource command unassigns one or more resources that have been assigned to the user's process by the Resource Control Package (RCP).

Usage

ur resources {-control\_args}

where:

1. resources  
specify the resources to be unassigned from the user's process. Currently, the only resources managed by RCP are devices. If a device is attached, it is automatically detached. A user can unassign all devices assigned to the process by specifying the -all control argument, or unassign one device by specifying its name.
2. control\_args  
can be chosen from the following:
  - comment STR, -com STR  
is a comment string that is displayed to the operator when the resource is unassigned. This comment is displayed only once, even if several resources are being unassigned. (See the assign\_resource command for details about comment strings.)
  - admin, -am  
forces an unassignment. This control argument should be specified by highly privileged users who want to unassign a resource that is assigned to some other process.
  - all, -a  
unassigns all devices assigned to the process.

Examples

In the example that follows, the user unassigns a tape previously assigned by the assign\_resource command by typing the command line:

```
! ur tape_03
```

## SECTION 5

### USER SUBROUTINES

This section contains descriptions of Multics subroutines, presented in alphabetical order. Each description contains the name of the subroutine, discusses the purpose of the subroutine, lists the entry points, and describes the correct usage for each entry point. Notes and examples are included when deemed necessary for clarity. The discussion below briefly describes the context of the various divisions of the subroutine descriptions.

#### Name

The "Name" heading shows the acceptable name by which the subroutine is called. The name is usually followed by a discussion of the purpose and function of the subroutine and the results that may be expected from calling it.

#### Entry

Each "Entry" heading lists an entry point of the subroutine call. This heading may or may not appear in a subroutine description; its use is entirely dependent upon the purpose and function of the individual subroutine.

#### Usage

This part of the subroutine description first shows the proper format to use when calling the subroutine and then explains each element of the call. Generally, the format is shown in two parts: a declare statement that gives the arguments in PL/I notation and a call line that gives an example of correct usage. Each argument of the call line is then explained. Arguments can be assumed to be required unless otherwise specified. Arguments that must be defined before calling the subroutine are identified as Input; those arguments defined by the subroutine are identified as Output.

#### Notes

Comments or clarifications that relate to the subroutine as a whole (or to an entry point) are given under the "Notes" heading.

## Other Headings

Additional headings are used in some descriptions, particularly the more lengthy ones, to introduce specific subject matter. These additional headings may appear in place of, or in addition to, the notes.

## Status Codes

The standard status codes returned by the subroutines are further identified, when appropriate, as either storage system or I/O system. For convenience, the most often encountered codes are listed in Appendix B of the MPM Subroutines. They are divided into three categories: storage system, I/O system, and other. Certain codes have been included in the individual subroutine description if they have a special meaning in the context of that subroutine. The reader should not assume that the code(s) given in a particular subroutine description are the only ones that can be returned.



Name: cv\_rcp\_attributes\_

The cv\_rcp\_attributes\_ subroutine contains several entry points that are useful in manipulating RCP resource attribute specifications and descriptions.

RCP resource attribute descriptions are printable strings that describe the attributes of resources (devices and volumes). For a description of the syntax of attribute descriptions see the Multics Administrators' Manual Project, Order No. AK51.

RCP resource attribute specifications are encoded representations of attribute descriptions. They may be either absolute, relative, or multiple. An absolute attribute specification represents a complete and consistent state of all the attributes of a resource. A relative attribute description represents a desired modification to the state of all the attributes of a resource, and must be applied to an absolute attribute specification to produce the desired change in that absolute specification. A multiple attribute specification does not represent a consistent state of all the attributes of a resource at any given time, but is useful for representing the union of all such consistent states, i.e., potential attributes.

---

Entry: cv\_rcp\_attributes\_\$to\_string

This entry point takes an RCP resource attribute specification and produces a printable RCP attribute description.

#### Usage

```
declare cv_rcp_attributes_$to_string entry (char (*), bit (72)
      dimension(2), char (*) varying, fixed bin (35));

call cv_rcp_attributes_$to_string (type, attributes, string, code);
```

where:

1. type (Input)  
specifies the type of resource from which attributes was obtained  
e.g., disk\_drive (see "Notes" below).
2. attributes (Input)  
is an RCP attribute specification (see "Notes" below).
3. string (Output)  
is a printable RCP attribute description.
4. code (Output)  
is a standard status code.

### Notes

A list of defined resource types may be obtained via the `list_resource_types` command.

---

Entry: `cv_rcp_attributes_$from_string`

This entry point accepts a printable RCP attribute description and produces an RCP attribute specification.

### Usage

```
declare cv_rcp_attributes $from_string entry (char (*), bit (72)
    dimension(2), char (*) varying, fixed bin (35));

call cv_rcp_attributes_$from_string (type, attributes, string, code);
```

where:

1. `type` (Input)  
specifies the type of resource to which attributes applies.
  2. `attributes` (Output)  
is the same as above.
  3. `string` (Input)  
is the same as above.
  4. `code` (Output)  
is the same as above.
- 

Entry: `cv_rcp_attributes_$modify`

This entry point applies a printable RCP resource attribute description (representing a relative attribute specification) to a given resource specification and returns a new attribute specification as the result. The resulting attribute specification consists of the original attribute specification, modified by the attributes specified in the printable description.

Usage

```
declare cv_rcp_attributes_$modify entry (char (*), bit (72) dimension (2),
char (*) varying, bit (72) dimension (2), fixed bin (35));

call cv_rcp_attributes_$modify (type, attributes, string, new_attributes,
code);
```

where:

1. type (Input)  
specifies the type of resource to which attributes and string apply.
2. attributes (Input)  
is an absolute RCP attribute specification.
3. string (Input)  
is a printable RCP attribute description that is to modify attributes.
4. new\_attributes (Output)  
is the new absolute RCP attribute specification.
5. code (Output)  
is the same as above.

---

Entry: cv\_rcp\_attributes\_\$from\_string\_rel

This entry point generates a relative attribute specification that can later be applied to attribute specifications of specific resources via the cv\_rcp\_attributes\_\$modify\_rel entry point.

Usage

```
declare cv_rcp_attributes_$from_string_rel entry (char (*),
char (*) varying, bit (72) dimension (4), fixed bin (35));

call cv_rcp_attributes_$from_string_rel (type, string, rel_attributes,
code);
```

where:

1. type (Input)  
specifies the type of resource to which string applies.
2. string (Input)  
is a printable RCP attribute description.
3. rel\_attributes (Output)  
is the relative RCP attribute specification.
4. code (Output)  
is the same as above.

---

Entry: cv\_rcp\_attributes\_\$modify\_rel

This entry point applies a relative attribute specification produced by the cv\_rcp\_attributes\_\$from\_string\_rel entry point to an absolute attribute specification of a specific resource.

#### Usage

```
declare cv_rcp_attributes_$modify_rel entry (bit (72) dimension (2),  
bit (72) dimension (4), bit (72) dimension (2));  
  
call cv_rcp_attributes_$modify_rel (attributes, rel_attributes,  
new_attributes);
```

where:

1. attributes (Input)  
is an absolute attribute specification.
2. rel\_attributes (Input)  
is a relative attribute specification to be applied to attributes.
3. new\_attributes (Output)  
is the resulting absolute attribute specification.

#### Notes

The caller must ensure that attributes and rel\_attributes refer to the same resource type, i.e., were generated by previous calls to cv\_rcp\_attributes\_ where the type arguments were identical.

---

Entry: cv\_rcp\_attributes\_\$reduce\_implications

This entry point accepts an attribute specification for a volume and returns the necessary minimal attribute specification that a device must possess to be able to accept the volume.

Usage

```
declare cv_rcp_attributes $reduce_implications entry (char (*), bit (72)
dimension(2), char (*), bit (72) dimension (4), fixed bin (35));

call cv_rcp_attributes_$reduce_implications (vol_type, vol_attributes,
dev_type, dev_attributes, code);
```

where:

1. vol\_type (Input)  
specifies the type of volume from which vol\_attributes was obtained.
2. vol\_attributes (Input)  
is an absolute attribute specification for the volume type specified.
3. dev\_type (Output)  
is the resource type of the device that accepts the given volume type.
4. dev\_attributes (Output)  
is a minimal relative attribute specification for a device capable of accepting a volume with the given attributes.
5. code (Output)  
is the same as above.

---

Entry: cv\_rcp\_attributes\_\$protected\_change

This function entry point accepts an absolute attribute specification for a resource and a relative attribute specification which is to modify it. It returns a value expressing whether or not this modification would affect protected attributes of the resource. No modification is actually attempted by this entry.

### Usage

```
declare cv_rcp_attributes_$protected_change entry (bit (72) dimension(2),
    bit (72) dimension(4)) returns (bit (1) aligned);

protected_change = cv_rcp_attributes_$protected_change (attributes,
    rel_attributes);
```

where:

1. attributes (Input)  
is an RCP attribute specification.
2. rel\_attributes (Input)  
is a relative attribute specification to be applied to attributes.
3. protected\_change (Output)  
is "1"b if this operation would modify protected attributes of the resource; otherwise, it is "0"b.

---

Entry: cv\_rcp\_attributes\_\$test\_valid

This entry point is used to determine whether a given attribute specification is absolute, relative, multiple, or invalid.

### Usage

```
declare cv_rcp_attributes_$test_valid entry (char(*), bit 72 dimension (2),
    fixed bin, fixed bin (35));

call cv_rcp_attributes_$test_valid (type, attributes, validity, code);
```

where:

1. type (Input)  
specifies the type of resource to which attributes applies.
2. attributes (Input)  
is an RCP attribute specification.
3. validity (Output)  
shows whether the attribute specification is absolute, relative, or multiple.  
0 is an absolute attribute specification  
1 is a relative attribute specification  
2 is a multiple attribute specification
4. code (Output)  
is a standard status code.

---

resource\_control\_

---

---

resource\_control\_

---

Name: resource\_control\_

The resource\_control\_ subroutine provides an interface to the Multics resource control facility. Entry points in this subroutine allow programs to reserve or cancel I/O devices and volumes.

---

Entry: resource\_control\_\$reserve

This entry point reserves a resource or group of resources for use by a process.

Usage

```
declare resource_control_$reserve entry (pointer, pointer, bit (1) aligned,  
bit (72) aligned, fixed bin (35));
```

```
call resource_control_$reserve (descriptions_ptr, reservation_desc_ptr,  
authorization, system, code);
```

where:

1. descriptions\_ptr (Input)  
is a pointer to the structure containing a description of the resources to be reserved (see "Resource Description" below).
2. reservation\_desc\_ptr (Input)  
is a pointer to the structure containing reservation information for the resources to be reserved (see "Reservation Description" below).
3. authorization (Input)  
checks the user's authorization to use the devices or volumes and is only valid if system = "1"b.
4. system (Input)  
specifies, if "1"b, that the calling process wishes to perform a privileged reservation (see "Notes" below).
5. code (Output)  
is a standard status code.

### Reservation Description

The reservation\_desc\_ptr argument points to the following structure (declared in the include file resource\_control\_desc.incl.pl1):

```

dcl 1 reservation_description aligned based,
    2 version_no fixed bin,
    2 reserved_for char (32),
    2 reserved_by char (32),
    2 reservation_id fixed bin (71),
    2 group_starting_time fixed bin (71),
    2 asap_duration fixed bin (71),
    2 flags aligned,
      (3 auto_expire bit (1),
       3 asap_bit (1),
       3 rel bit (1),
       3 sec bit (1)) unaligned,
    2 n_items fixed bin,
    2 reservation_group (Resource_count refer
      (reservation_description.n_items)),
    3 starting_time fixed bin (71),
    3 duration fixed bin (71);
```

where:

1. version\_no (Input)  
is the current version number of this structure. It should be set to "resource\_control\_version\_1".
2. reserved\_for (Input)  
specifies the User\_id of the process for whom this reservation is made. The use of an asterisk (\*) for a component name is permitted. If this element is blanks, the User\_id of the current process is used.
3. reserved\_by (Input)  
is the User\_id of the process which is charged for this reservation (see "Notes" below). This element is ignored for an unprivileged reservation and the current User\_id is used.
4. reservation\_id (Input or Output)  
is an identifier for this reservation group. It is currently returned as an absolute clock time.
5. n\_items (Input)  
is the number of items being reserved.

The rest of the items in this structure are currently ignored and should be set to zero.

### Notes

If system = "1"b, reservation\_description.reserved\_by is used to specify the User\_id of the process to be charged for this reservation.



---

resource\_control\_

---

---

resource\_control\_

---

The reservation\_description structure is strongly dependent on the resource\_descriptions structure. That is, for each resource described in resource\_descriptions there must be a corresponding entry of the same index in reservation\_description.

### Access Restrictions

Execute access to the rep\_sys\_ gate is necessary to perform a privileged reservation.

---

Entry: resource\_control\_\$cancel\_id\_string

This entry point cancels the reservation of a resource or group of resources.

### Usage

```
declare resource_control_$cancel_id_string entry (char(*), char(*),
          bit(1) aligned, fixed bin (35));

call resource_control_$cancel_id_string (reservation_id, group_id, system,
          code);
```

where:

1. reservation\_id (Input)  
is the character string representation of the reservation identifier to be cancelled.
2. group\_id (Input)  
is the group id of the user to whom the reservation belongs. This is only valid if system = "1"b.
3. system (Input)  
specifies, if "1"b, that a privileged cancellation is to be performed (see "Notes" below).
4. code (Output)  
is a standard status code.

### Notes

If system = "1"b, then the reservation group is forcibly cancelled whether or not it belongs to the current process.

### Access Restrictions

Execute access to the rcp\_sys\_ gate is necessary to perform a privileged cancellation.

### Resource Description

The descriptions\_ptr argument points to the following structure (this structure is declared in the include file resource\_control\_desc.incl.pl1):

```
dcl 1 resource_descriptions based (resource_desc_ptr) aligned,
  2 version_no fixed bin,
  2 n_items fixed bin,
  2 item (Resource_count refer (resource_descriptions.n_items)) aligned,
    3 type char (32),
    3 name char (32),
    3 uid bit (36),
    3 potential_attributes bit (72),
    3 attributes (2) bit (72),
    3 desired_attributes (4) bit (72),
    3 potential_aim_range (2) bit (72),
    3 aim_range (2) bit (72),
    3 owner char (32),
    3 acs_path char (168),
    3 location char (168),
    3 comment char (168),
    3 charge_type char (32),
    3 rew bit (3) unaligned,
    3 (usage_lock,
      release_lock,
      awaiting_clear,
      user_alloc) bit (1) unaligned,
    3 pad2 bit (29) unaligned,
    3 given aligned,
      4 (name,
        uid,
        potential_attributes,
        desired_attributes,
        potential_aim_range,
        aim_range,
        owner,
        acs_path,
        location,
        comment,
        charge_type,
        usage_lock,
        release_lock,
        user_alloc) bit (1),
      4 pad1 bit (22)) unaligned,
    3 state bit (36) aligned,
    3 status_code fixed bin (35);
```

where:

1. `version_no` (Input)  
is the current version number of the structure. It should be set to "resource\_control\_version\_1".
2. `n_items` (Input)  
specifies the number of resources described by this structure. A consistent combination of the following elements must be supplied for each resource described.
3. `type` (Input)  
specifies the type of resource desired (e.g., tape, disk\_drive). It must be supplied (see "Notes" below).
4. `name` (Input or Output)  
is a specific resource name. If `flags.name_given = "1"`, the named resource is chosen. If `flags.name_given = "0"`, a resource is chosen depending on criteria specified by other elements of the structure, and the name of the resource chosen is returned in this element (see "Notes" below).
5. `uid` (Input or Output)  
is the unique identifier of a specific resource. If `flags.uid_given = "1"`, the specified resource is chosen. If `flags.uid_given = "0"`, a resource is chosen depending on criteria specified by other elements of the structure, and the unique identifier of the resource chosen is returned in this element.
6. `potential_attributes` (Output)  
specifies the potential attributes of the resource chosen.
7. `attributes` (Input or Output)  
contains, if `flags.attr_given = "1"`, the specification of attributes which the resource chosen must possess. If `flags.attr_given = "0"`, the resource to be chosen need not possess any particular attributes. The attributes of the resource chosen are returned in these elements (see "Notes" below).
8. `desired_attributes` (Input)  
specifies the desired attributes of the resource chosen.
9. `potential_aim_bounds` (Output)  
are a pair of AIM access classes, specifying the minimum and maximum process authorization that can be permitted to acquire this resource.
10. `aim_bounds` (Input or Output)  
are a pair of AIM access classes, specifying the minimum and maximum process authorization that can be permitted to both read and write this resource. If `flags.aim_bounds_given = "1"`, this element is input. Otherwise, it is output.
11. `owner` (Input or Output)  
is the owner of the resource. If `flags.owner = "1"`, this element is input. Otherwise, this element is output (see "Notes" and "Access Restrictions" below).
12. `acs_path` (Input)  
is the pathname of the access control segment (ACS) for this resource (see "Access Restrictions" below).

13. location (Output)  
contains a character string description of the location of this resource.
14. comment (Input)  
contains a character-string comment which is associated with this resource.
15. charge\_type (Input)  
is the accounting identifier for this resource.
16. rew (Output)  
is the effective access of the user to this resource.
17. usage\_lock (Input)  
if "1"b, specifies that this resource cannot be used by any user, regardless of the state of the resource.
18. release\_lock (Input)  
if "1"b, specifies that the owner of the resource is not allowed to release the resource. Unless system = "1"b, this element is ignored (see "Notes" below).
19. awaiting\_clear (Output)  
specifies that the resource is awaiting manual clear.
20. user\_alloc (Input)  
if "1"b, specifies that the user has not allocated the resource to any use.
21. pad2 (Input)  
is unused and must be zero.
22. name (Input)  
is "1"b if item.name has been supplied by the caller.
23. uid (Input)  
is "1"b if item.uid has been supplied by the caller.
24. potential\_attr (Input)  
is "1"b, if item.potential\_attributes has been supplied by the caller.
25. desired\_attr (Input)  
is "1"b if item.desired\_attributes has been supplied by the caller.
26. potential\_aim\_bounds (Input)  
is "1"b if item.potential\_aim\_bounds has been supplied by the caller.
27. aim\_bounds (Input)  
is "1"b if item.aim\_bounds has been supplied by the caller.
28. owner (Input)  
is "1"b if item.owner has been supplied by the caller.
29. acs\_path (Input)  
is "1"b if item.acs\_path has been supplied by the caller.
30. location (Input)  
is "1"b if item.location has been supplied by the caller.

31. comment (Input)  
is "1"b if item.comment has been supplied by the caller.
32. charge\_type (Input)  
is "1"b if item.charge\_type\_given has been supplied by the caller.
33. usage\_lock (Input)  
is "1"b if item.usage\_lock has been supplied by the caller.
34. release\_lock (Input)  
is "1"b if item.release\_lock has been supplied by the caller.
35. user\_alloc (Input)  
is "1"b if item.user\_alloc\_given has been supplied by the caller.
36. pad1 (Input)  
is unused and must be zero.
37. state (Output)  
is for the use of resource\_control\_ and should not be used by the user.
38. status\_code (Output)  
is a standard status code. If the subroutine argument code is nonzero, one or more items in the structure have a nonzero status\_code specifying in more detail why the attempt to manipulate the described resource was refused.

### Notes

A list of defined resource types may be obtained via the list\_resource\_types command.

Suitable values for the attributes element may be constructed using the cv\_rcp\_attributes\_\$from\_string subroutine.

### Access Restrictions

The user must have at least sm permission to the directory in which the ACS is specified to reside.

Unless otherwise stated, the user must have re access to the rcp\_sys\_gate to specify system = "1"b in the calling sequence for any entry point of the resource\_control\_ subroutine.

---

resource\_info\_

---

---

resource\_info\_

---

Name: resource\_info\_

The resource\_info\_ subroutine returns selected information about RCP resource types defined on the system.

---

Entry: resource\_info\_\$get\_type

This entry point, given the name of a resource type, indicates whether the resource type named is a device or a volume.

Usage

```
declare resource_info_$get_type entry (char (*), bit (1), fixed bin (35));
call resource_info_$get_type (name, is_volume, code);
```

where:

1. name (Input)  
is the name of a defined resource type (see "Notes" below).
2. is\_volume (Output)  
is "1"b if the resource type given specifies a class of volumes. If "0"b, the resource type given specifies a class of devices.
3. code (Output)  
is a standard status code.

Notes

A list of defined resource types may be obtained via the list\_resource\_types command (see Section 4).

---

Entry: resource\_info\_\$limits

This entry point returns information about quantity and time limits for a given resource type.

---

resource\_info\_

---

---

resource\_info\_

---

### Usage

```
declare resource_info $limits entry (char (*), fixed bin, fixed bin,  
    fixed bin, fixed bin (35));
```

```
call resource_info $limits (name, max_quantity, default_time, max_time,  
    code);
```

where:

1. name (Input)  
is the name of a defined resource type.
2. max\_quantity (Output)  
is the maximum number of this type of resource that a process may  
assign at one time.
3. default\_time (Output)  
is the default reservation time, in minutes, for this type of  
resource.
4. max\_time (Output)  
is the maximum allowed reservation time, in minutes, for this type  
of resource.
5. code (Output)  
is a standard status code.

### Notes

The information returned by this entry point is from the RTDT. These are not the limits currently enforced by RCP (see "Device Limits" in Section 1).

---

Entry: resource\_info \$mates

This entry provides information about the resource type or types with which the given resource type may be mounted.

### Usage

```
declare resource_info $mates entry (char (*), fixed bin, char (*),  
    dimension (*), fixed bin (35));
```

```
call resource_info $mates (name, n_mates, mates, code);
```

where:

1. name (Input)  
is the name of a defined resource type.
2. n\_mates (Output)  
is the number of mates returned.
3. mates (Output)  
contains the name or names of the resource type(s) that may be mounted with this resource (see "Notes" below).
4. code (Output)  
is a standard status code.

### Notes

If the number of elements in mates is too small to hold all the mates for the given resource type, code is set to error\_table\_\$smallarg and mates is set to the null string. However, n\_mates still contains the number of mates associated with the given resource type.

---

Entry: resource\_info\_\$defaults

This entry point fills a resource\_descriptions structure with the default registration parameters defined in the RTDT.

### Usage

```
dcl resource_info_$defaults entry (char(*), char(*), pointer,  
fixed bin(35));  
call resource_info_$defaults (name, subtype, item_ptr, code);
```

where:

1. name (Input)  
is the name of a defined resource type.
2. subtype (Input)  
is the name of a subtype of the resource type, defined in the RTDT. If subtype is the null string, the master defaults for the resource type are used.
3. item\_ptr  
points to a structure declared like resource\_descriptions.item (see the resource\_control\_ subroutine).
4. code (Output)  
is a standard status code.



---

resource\_info\_

---

---

resource\_info\_

---

Entry: resource\_info\_\$lock\_on\_release

This entry point returns a value specifying whether resources of a given type are to be locked for manual clearing at release time.

Usage

```
dcl resource_info_$lock_on_release entry (char(*), bit(1) aligned,  
    fixed bin(35));
```

```
call resource_info_$lock_on_release (name, lock_sw, code);
```

where:

1. name (Input)  
is the name of a defined resource type.
2. lock\_sw (Output)  
specifies whether the resource is locked at release time.  
"1"b lock the resource  
"0"b do not lock the resource
3. code (Output)  
is a standard status code.

---

Entry: resource\_info\_\$canonicalize\_name

This entry point applies the proper canonicalization to a resource name of a given resource type. See "Canonicalization Routines" in the MAM RCP.

Usage

```
declare resource_info_$canonicalize_name entry (char(*), char(*), char(*),  
    fixed bin(35));
```

```
call resource_info_$canonicalize_name (resource_type, resource_name,  
    canonicalized_name, code);
```

---

resource\_info\_

---

---

resource\_info\_

---

where:

1. resource\_type (Input)  
is the name of a defined resource type.
2. resource\_name (Input)  
is the string to be canonicalized.
3. canonicalized\_name (Output)  
is the canonicalized representation of resource\_name.
4. code (Output)  
is a standard status code.

## INDEX

- A
- access control
  - resources 1-1, 1-7
  - access control segment (ACS) 1-2
- access control segment (ACS) 1-2, 1-7
- access class ranges 1-8
  - manipulating RCP effective access 1-9
  - RCP effective access 1-8
- acquire\_resource (aqr) command 4-4
- acquisition 1-4, 1-10
- administrative and user interfaces 3-1
  - administrative 3-1
  - user 3-1
    - acquiring and using resources 3-1
    - user manipulation of registries 3-2
- aqr
  - see acquire\_resource command
- ar
  - see assign\_resource command
- assigning devices 1-5
- assign\_resource (ar) command 4-7
- attaching devices 1-6
- C
- cancel\_resource (cnr) command 4-10
- cnr
  - see cancel\_resource command
- cv\_rep\_attributes\_subroutine 5-3
- D
- device
  - limits
    - workspace size 1-10
  - names 1-6
- G
- glossary 1-12
- I
- I/O
  - attachments
    - list\_resources 4-12
- I/O (cont)
  - peripheral
    - assign\_resource 4-7
    - reserve\_resource 4-16
    - unassign\_resource 4-22
- L
- limits, devices
  - workspace size 1-10
- list\_resources (lr) command 4-12
- list\_resource\_types (lrt) command 4-11
- logical volume
  - list\_resources 4-12
- lr
  - see list\_resources command
- lrt
  - see list\_resource\_types command
- R
- RCP
  - see resource control package (RCP)
- RCP effective access 1-8
- release\_resource (rlr) command 4-15
- reservation
  - cancel\_resource command 4-10
- reservation identifier
  - cancel\_resource command 4-10
- reserve\_resource (rsr) command 4-16
- reserving resources 1-4
- resource control package (RCP) 1-1
  - functions
    - access control
      - resources 1-1, 1-2, 1-7
    - assigning devices 1-2, 1-5
    - attaching devices 1-2, 1-6
    - cancelling resources 1-2
    - detaching devices 1-2
    - device control functions 1-2
    - reserving resources 1-2, 1-4
    - resource information 1-2
    - unassigning devices 1-2
    - RCP effective access 1-8
    - resource types
      - resource\_info\_ 5-16
- resource information 1-2
- resource management 1-1

resource type master file  
  syntax of the RIMF  
  naming rules for attributes 3-2

resource\_control\_subroutine 5-9

resource\_info\_subroutine 5-16

resource\_status (rst) command 4-18

rlr  
  see release\_resource command

rsr  
  see reserve\_resource command

rst  
  see resource\_status command

## S

scratch 1-7

setr  
  see set\_resource command

set\_resource (setr) command 4-20

system pool 1-10

## T

tape  
  access control 2-2  
  acquisition 2-1  
  registration 2-1

## U

unassign\_resource (ur) command 4-22

ur  
  see unassign\_resource command

user commands 4-1

user subroutines 5-1

user\_id 1-14

## V

volume names 1-7  
  reserved  
  scratch 1-7  
  T&D\_Volume 1-7

## W

workspace size 1-10

HONEYWELL INFORMATION SYSTEMS  
Technical Publications Remarks Form

CUT ALONG LINE

TITLE LEVEL 68  
MULTICS RESOURCE CONTROL USERS' GUIDE

ORDER NO. CT38-00

DATED JUNE 1981

ERRORS IN PUBLICATION

Large empty rectangular box for reporting errors in the publication.

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Large empty rectangular box for providing suggestions for improvement to the publication.



Your comments will be investigated by appropriate technical personnel and action will be taken as required. Receipt of all forms will be acknowledged; however, if you require a detailed reply, check here.

FROM: NAME \_\_\_\_\_

DATE \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY \_\_\_\_\_

ADDRESS \_\_\_\_\_

\_\_\_\_\_

PLEASE FOLD AND TAPE—  
NOTE: U. S. Postal Service will not deliver stapled forms



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 39531 WALTHAM, MA02154

POSTAGE WILL BE PAID BY ADDRESSEE

**HONEYWELL INFORMATION SYSTEMS**  
200 SMITH STREET  
WALTHAM, MA 02154

ATTN: PUBLICATIONS, MS486



CUT ALONG LINE  
FOLD ALONG LINE  
FOLD ALONG LINE

**Honeywell**

# Honeywell

## Honeywell Information Systems

In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154  
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5  
In the U.K.: Great West Road, Brentford, Middlesex TW8 9DH  
In Australia: 124 Walker Street, North Sydney, N.S.W. 2060  
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

31942, 7.5C781, Printed in U.S.A.

CT38-00