VM Batch Facility

Installation, Customization, and Administration

Version 2 Release 2

VM Batch Facility

Installation, Customization, and Administration

Version 2 Release 2

┌─ **Note!** ──┐

Before using this information and the product it supports, read the information in "Notices" on page vii.

└──┘

Third Edition (July 1999)

This edition applies to Version 2, Release 2, Modification 0 of IBM® Batch Facility (product number 5684-137) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC24-5573-01.

Contents

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes to the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Mail Station P300,
522 South Road
Poughkeepsie, NY 12601-5400
U.S.A.
Attention: Information Request
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities on non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM

for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Programming Interface Information

This book is intended to help the customer to do the installation of the VM Batch Facility. It contains information on installing, customizing, and administering this facility.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

BookManager	IBM
DFSMS/VM	IBMLink
Language Environment	Library Reader
OpenEdition	RACF
S/390	VM/ESA

Other company, product, and service names may be trademarks or service marks of others.

Preface

This Book Explains How To

- Install the VM Batch Facility in VM systems.

- Customize the VM Batch Facility.

- Perform VM Batch Facility operation and administration tasks and use authorized user commands.

Who Should Use This Book

- Installers of the VM Batch Facility.

- Customizers of the VM Batch Facility.

- Administrators and operators of the VM Batch Facility who need information on performing operation and administration tasks.

- Any authorized user of the VM Batch Facility who needs information on authorized user commands and tasks.

Other Publications

The VM Batch Facility *User's Guide*, SC24-5574-02, explains how to use the VM Batch Facility to process jobs.

The VM Batch Facility *General Information* manual, GC24-5572-01, and the VM Batch Facility *Licensed Program Specifications*, GC24-5571-01, provide an overview of the VM Batch Facility and planning information.

The VM Batch Facility binder label, SX66-0206, identifies the VM Batch Facility manuals if you are storing them in a VM ring binder, SX24-5129.

All publications, as well as the label and binder, can be ordered with a single Bill of Forms, SBOF-0252.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other VM/ESA documentation:

- Visit our home page at:

  http://www.ibm.com/s390/vm/related/vmbatch

  There you will find the feedback page where you can enter and submit your comments.

- Send your comments by electronic mail to one of the following addresses:

Format	Address
Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL
IBMLink™	GDLVME(PUBRCF)

Be sure to include the name of the book, the form number (including the suffix), and the page, section title, or topic you are commenting on.

- Fill out the form at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

Summary of Changes

This section lists the changes to VM Batch Facility in Version 2 Release 2 that apply to the administrator. Additional important changes are listed in the beginning of the *VM Batch Facility User's Guide*.

1. **Job spawning**

   You can enable a monitor to accept jobs and commands that have been issued from within a VM Batch Facility job.

2. **Vector support for accounting records**

   CP accounting information for Vector Facility Processors is made available to the Job Accounting Exit.

3. **Specification of storage requirements for jobs**

   You can define the range of permissable virtual storage sizes for jobs of a class.

4. **Optional Passwords for class AUTOLOG jobs**

   You can configure your VM Batch Facility to require or not require passwords for class AUTOLOG jobs. This helps to reduce the occurrences where CP prevents the VM Batch Facility monitor from auto-logging user and task machines when the threshold for incorrect passwords has been reached.

5. **Usability Exits**

   Two new usability exits, the user interface installation exit (DGREVR EXEC) and the task control installation exit (DGROOLY EXEC), increase your ability to tailor the VM Batch Facility to your particular needs.

6. **Installation option to save job to disk**

   You have the option to choose whether incoming jobs are stored on the monitor's 193 disk or transferred directly from the monitor's virtual punch to the task machine for execution.

7. **BATCH LEVEL command**

   This new command provides the facility for service personnel to interrogate the service levels of each serviceable component of the VM Batch Facility.

8. **Scheduling of recurring jobs**

   You can configure the VM Batch Facility to allow jobs to be submitted for repeated execution, without requiring the user to resubmit the jobs on each occasion.

9. **JOBSTEP for DGRUJB**

   The VM Batch Facility provides for a call to be made to the job exit DGRUJB EXEC at the completion of each jobstep, to provide you with more granular control of job operations.

   The sample DGRUJB EXEC provided with this release issues a message to a job owner notifying that the job *has* started when called for the first jobstep; rather than notifying that the job *is to be* started when selected for dispatch.

10. **Dynamic reload of installation exits**

   Installation exits running on a monitor can be refreshed by using a command, without the need to shut the monitor down.

11. **Fast CANCEL**

   The cancellation process is sped up significantly for all jobs, especially if the ERROR option is NODUMP.

12. **CONSOLE option for SUBMIT command**

   Job submittors can request at the time of submission that no console be spooled from their VM Batch Facility job.

13. **Submission of jobs to CLASS with no task machine**

   The VM Batch Facility allows jobs to be submitted for a class that has no task machines associated with it.

14. **Rationalization of Return codes and severity codes**

   The conventions for return codes from RUNBATCH and the VM Batch Facility commands, and the severity codes for messages, have been established and are now documented.

15. **Secure parameter added to TASK keyword in CONTROL FILE**

   The secure parameter has been added to the TASK statement. For both Format-1 and Format-2 TASK entries, it follows the end time. For Format-2 entries is precedes the password3 entry.

Chapter 1. Introduction

The VM Batch Facility is a batch processing program. Its function is to perform computer jobs on behalf of computer users, effectively freeing them to work at their terminals while their jobs run elsewhere. It can also be exploited to balance a system's work load by shifting work to off-peak times, when the system is not busy.

Overview

The control program of a VM Batch Facility runs in a disconnected service virtual machine, called a VM Batch Facility *monitor*.

The monitor manages secondary service virtual machines, called VM Batch Facility *task* machines, to execute jobs on behalf of users. Task machines are logged on by a monitor when needed to run a job, and log off when their job is done.

A VM Batch Facility job consists of a REXX exec, prepared by a user, and submitted to a VM Batch Facility monitor for execution. When a job is submitted, the user may include *job control options*, that specify how the job is to be managed by the monitor; for example, when the job may run, or how much virtual storage is to be provided.

Each job submitted is assigned a job *class*. Classes are used to group jobs with similar operating characteristics, and are a convenient means of enforcing limits on job control options.

Administrators are users who have been granted authority to perform privileged functions. Administrators may influence the execution of jobs belonging to other users (but not the jobs themselves), and may issue privileged VM Batch Facility commands such as the termination of a monitor.

The *alertee* is an administrator selected to receive *alert messages* when an unusual situation develops.

The configuration of a monitor is defined in a *CONTROL FILE*. The CONTROL FILE defines the task machines that are available to the monitor, the classes that it supports, which users are designated as administrators, and various operational parameters.

At strategic points in the processing of a job or command, installation exits are called to apply installation criteria to processing decisions; for example, to restrict the use of a job class to selected users.

When a user submits a job, the user interface of the VM Batch Facility runs in the user's virtual machine to perform

- preliminary command validation such as syntax checking,

- composition of the job and the selected job control options into a format acceptable by a monitor,

- management of the transfer of the job to the chosen monitor.

When the job is received by a monitor:

- A control block is allocated to the job. Each job is uniquely identified by the triplet made up of the user ID of the submittor, the ID of the owner's node, and the name given to the job by the submittor. If an active job with same ID already exists then the job is rejected.

- Validation is performed in the monitor context; for example, checking that the levels of resource requested for the job do not conflict with the defined class limits.

- The job is subjected to installation exit screening.

- The job is stored on disk or in the monitor's virtual punch.

- If necessary, the job waits until a task machine is available with which it is compatible (according to the task definition in the CONTROL FILE).

- It competes with other ready and compatible jobs for the task machine, according to its priority.

- The task machine's 191 minidisk is cleaned.

- When selected, dispatch is verified by installation exits.

- The job is transferred to the task machine's virtual reader, and the task machine is autologged to run the job.

When the task machine is autologged, the task interface of the VM Batch Facility is invoked in the task virtual machine, to initialize the machine and read the job onto disk. The task interface then invokes the job as a REXX exec.

While the job is running, the monitor

- Records the resources used by the job.

- Checks that the task machine does not consume system resources in excess of the limits defined for the job.

- Checks that the task machine does not become inactive without the job terminating.

When the job ends

- The task machine 191 disk is erased and any remaining spool files are purged.

- Installation exits are notified that the job has ended. In particular, the monitor job accounting exit may be called to record the amount of system resource used by the job.

- If the job recurs then it rescheduled to run again, otherwise it is discarded.

Some features of the VM Batch Facility are listed below:

Autolog jobs
A monitor may be enabled to run jobs on submitting users' own virtual machines.

Job scheduling
A job may be defined to have a start window, such that it will not become eligible to run until a specified date has arrived, and is then only eligible to run between the window start and end times.

Job chaining

A job may be defined to have a chain condition on another job, such that it will not become eligible to run until the job that it is chained to has completed or reached a particular jobstep.

Job status

Status information is available for all active jobs.

Alternate user ID

An alternate user ID may be set for a task machine for running a job.

Jobsteps

Running jobs may request that the monitor record a jobstep by name.

Job restart

After a system outage, running jobs may be restarted from the last notified jobstep.

Job consoles

The virtual console from a job is available to the job owner at job completion.

Error processing

A job may have a secondary processing routine defined, so that if the job is cancelled while running, the secondary routine is invoked to perform error recovery.

Job spawning

VM Batch Facility offers a limited capability for jobs to be submitted from within a job, and for such *spawned* jobs to be responsive to other spawned commands.

Job recurrence

VM Batch Facility offers an ability to have jobs repeat automatically according to installation defined *recurrence patterns*.

Queue manipulation

Commands are available to allow administrators to override normal scheduling mechanisms. For example,

- job priorities may be altered,

- queued jobs may be held, running jobs may be suspended,

- jobs may be forcibly dispatched, regardless of their scheduling constraints.

Monitor console

The virtual console of the monitor virtual machine records all activity at the monitor.

Load Level Scheduling

Automatic load level scheduling is available to maintain contention for system resources at acceptable levels.

Monitor networks

Multiple VM Batch Facility monitors may be run on a single system, and a monitor may accept jobs from other systems. Users are able to direct their jobs to an appropriate monitor.

Dynamic reconfiguration

There is no need to terminate a VM Batch Facility monitor in order to introduce a change to the CONTROL FILE or to refresh the monitor installation exits.

Direct interface

The VM Batch Facility monitors offer a limited direct interface, whereby jobs and commands may be accepted that have not been pre-processed by the VM Batch Facility user interface; for example when it is necessary to submit jobs from non VM systems.

Security

The VM Batch Facility enforces security in two ways:

1. Job owner passwords for autolog jobs are stored in encrypted form and are erased when no longer needed.

2. Jobs are kept separate from each other.

   The VM Batch Facility takes the actions listed below to prevent one job from gaining access to job files or data belonging to another job. When a job finishes, the VM Batch Facility:

   a. Erases the contents of the task machine 191 minidisk to destroy any job owner files sent to the task machine as part of a job. The 191 minidisk is erased and cleaned both before a job is started and after a job ends.

   b. Purges the printer, punch, and reader queues of the task machine. These actions are performed both after a job ends and before a new job is started to destroy any spooled input or output files that have not been closed or processed. (Exception: not performed at job completion if a job is ended by a BATCH FORCE or BATCH STOP JOBS command.)

   c. Logs off the task machine to detach minidisks linked by a job.

The VM Batch Facility uses DIAGNOSE X'D4' to assign an alternate user ID to a task machine. This is usually the job owner's user ID. Any spool files generated by the job belong to the alternate user ID, not to the task machine. Spool files cannot be obtained by a subsequent job submitted by another user running in the same task machine.

However, when the VM Batch Facility is running in an environment where DIAGNOSE X'D4' has been disabled, or where a DIAGNOSE X'D4' has not been issued against a task machine, due to a requested alternate user ID not being confirmed when confirmation was required, jobs can potentially obtain spool files created by previous jobs that ran in the same task machine, by the CP TRANSFER command. When no alternate user ID has been set for a task machine, spool files belong to the task machine itself. In such cases, users are advised not to spool job output, but to use other techniques. A sample exec in the *VM Batch Facility User's Guide* demonstrates how to return job output to a job owner without using spool files.

Restrictions on Using the VM Batch Facility

1. Jobs that use the Virtual Machine Communication Facility (VMCF), or modify the external new PSW, will have the SUSPEND and RESUME functions disabled for the duration of those jobs. In addition, the time taken to CANCEL these jobs is increased.

When Load Level Scheduling (LLS) is active, these jobs will not respond correctly to monitor control, and may be permanently suspended. They should be defined as independent of the LLS.

2. Jobs should not alter the CMS environment under which they run.

3. Programs which execute in the CMS user area in the user's virtual machine, (origin x'20000') cannot issue VM Batch Facility commands unless the DGRUVM MODULE has been loaded as a CMS nucleus extension. See "Specifying an automatic NUCXLOAD of DGRUVM MODULE" on page 163.

4. Users permitted to submit jobs to the VM Batch Facility who have user ID's which match certain VM Batch Facility command keywords will cause commands to malfunction, when commands are issued from those user ID's, and when they are used by other users to identify job ownership. Similarly, the node ID's on which VM Batch Facility users reside should not conflict with VM Batch Facility command keywords.

   The following keywords should not be permitted as user ID's or node ID's of VM Batch Facility users:

   - ACTIVE
   - ALL
   - CLEAR
   - CLOSE
   - HOLD
   - JOB
   - OLD
   - QUEUED
   - REPEATS
   - RUNNING
   - START
   - STOP
   - WAIT

5. Users of the VM Batch Facility that have an alternate user ID set for their virtual machine that is different from their user ID, will not be able to control jobs submitted to the VM Batch Facility. An exception is when VM Batch commands are issued from a VM Batch Facility task machine using the documented support for job spawning.

6. The string *NOPASS* is reserved as a keyword that indicates a null password. This string cannot be specified to CP or RACF to be–

   - a valid logon password for any VM Batch Facility user submitting a CLASS AUTOLOG job
   - a valid logon password for any VM Batch Facility task machine
   - a valid write password for any VM Batch Facility task machine 191 minidisk.

Special System Requirements

It is advisable to exclude the 191 minidisks of VM Batch Facility task machines from all system backup procedures. Failure to do so may result in task machines being marked OFFLINE, if the VM Batch Facility attempts to start a job on a task machine while its 191 disk is being backed up. There is no need to backup the task machine minidisks because the task machine disks serve only as work areas, and files are not preserved across jobs.

How to Read the Syntax Diagrams

Throughout this document, command syntax is described using the structure defined below.

- **Flow:**
  Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following are symbols used in the syntax diagrams:

  ►►——   Indicates the beginning of a syntax diagram.
  ——►   Indicates that the syntax diagram continues on the next line.
  ►——   Indicates that a syntax diagram continues from the previous line.
  ——►◄   Indicates the end of a syntax diagram.

- **Abbreviations:**
  Uppercase letters denote the shortest acceptable abbreviation. If an item appears entirely in uppercase letters, it cannot be abbreviated. You can type the item in uppercase letters, lowercase letters, or any combination.

  In this example, you can enter KEYWO, KEYWOR, or KEYWORD in any combination of uppercase and lowercase letters.

  ►►—KEYWOrd—►◄

- **Symbols:**
  You must code the following symbols exactly as they appear in the syntax diagram.

  *   Asterisk
  :   Colon
  ,   Comma
  =   Equal Sign
  -   Hyphen
  () Parentheses
  .   Period

- **Variables:**
  Highlighted lowercase items (*like this*) denote variables. In this example, *var_name* represents a variable you must specify when you code the KEYWORD command.

  ►►—KEYWOrd—*var_name*—►◄

- **Repetition:**
  An arrow returning to the left means that the item can be repeated.

  ►►—┬*repeat*┬—►◄

A character within the arrow means you must separate repeated items with that character.

```
►►──┬──,──────┬──►◄
    └─repeat──┘
```

A footnote (1) by the arrow references a limit that tells how many times the item can be repeated.

```
►►──┬──────────┬─(1)──►◄
    └─repeat───┘
```

Note:

1 Specify *repeat* up to 5 times.

- **Required Choices:**
  When two or more items are in a stack and one of them is on the line, you *must* specify one item. In this example, you must choose A, B, or C.

```
►►──┬──A──┬──►◄
    ├──B──┤
    └──C──┘
```
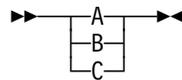
- **Optional Choice:**
  When an item is below the line, the item is optional. In this example, you can choose A or nothing at all.

```
►►──┬──────┬──►◄
    └──A───┘
```

  When two or more items are in a stack below the line, all of them are optional. In this example, you can choose A, B, C, or nothing at all.

```
►►──┬──────┬──►◄
    ├──A───┤
    ├──B───┤
    └──C───┘
```
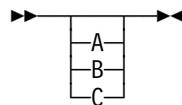
- **Defaults:**
  Defaults are above the line. The system uses the default unless you override it. You can override the default by coding an option from the stack below the line. In this example, A is the default. You can override A by choosing B or C.

```
    ┌──A──┐
►►──┼─────┼──►◄
    ├──B──┤
    └──C──┘
```
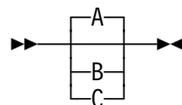
- **Repeatable Choices:**
  A stack of items followed by an arrow returning to the left means that you can select more than one item or, in some cases, repeat a single item. In this example, you can choose any combination of A, B, or C.

```
    ┌─────────┐
►►──┴──┬──A──┬─┴──►◄
       ├──B──┤
       └──C──┘
```
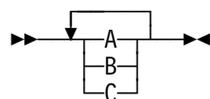
- **Syntax Fragments:**

  Some diagrams, because of their length, must fragment the syntax. The fragment name appears between vertical bars in the diagram. The expanded fragment appears in the diagram after a heading with the same fragment name. In this example, the fragment is named "A Fragment."

  ►►──┤ A Fragment ├──►◄

  A Fragment:

  ```
        ┌─A─┐
  ├──┼─B─┼──┤
        └─C─┘
  ```
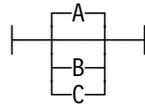
Chapter 2. Installation Notes

Detailed instructions on how to install the VM Batch Facility are provided in the *VM Batch Facility Program Directory* accompanying the machine readable material. The notes in this chapter assist in the installation and maintenance of the VM Batch Facility

Follow these steps when installing the VM Batch Facility:

1. Review the notes in this chapter that are relevant to your installation.

2. Install the program material according to the instructions in the *VM Batch Facility Program Directory*.

Migration Considerations

If you are migrating from a previous release of the VM Batch Facility, you should consider the following:

Monitor cold start

The first activation of VM Batch Facility Version 2 Release 2 requires a cold start of each monitor. Ensure that all jobs on your current VM Batch Facility have completed before installing the new version.

You may effect a cold start manually as described on page 73. Do not attempt to simulate a warm start by restoring the monitor work files from an earlier release, as these files are incompatible between releases.

VMSES/E enablement

VMSES/E has been implemented as the vehicle for installing and servicing the VM Batch Facility.

Refer to the *VM Batch Facility Program Directory* for instruction regarding the installation and application of service to the VM Batch Facility.

Monitor CONTROL FILE changes

You may be affected by the following changes to the processing of the CONTROL FILE:

1. All CLASS keywords must be in the format introduced with Version 2 Release 1, and described in this document.

2. The VSTOR parameter of the CLASS keyword has been replaced with the new DEFSTOR parameter. See "Filling Out the CLASS Keyword" on page 35.

3. The RACF keyword is no longer required. If it is included, a warning message is issued to the monitor console:

   ```
   DGRINI020W Invalid CONTROL FILE record ignored
   ```

4. If you intend to exploit the new ability to insist on logon passwords, ensure that your TASK keywords do not specify *NOPASS* as logon passwords.

Installation exits

The sample exits provided with the product, and the interfaces to all exits have changed. These are described in detail in "Migrating Exits" on page 11.

The messages in the supplied sample DGRUJB exit are issued at a different stage in job processing, from Version 2 Release 1. In particular the 'Job *jobid* has started' message is issued after the initial JOBSTEP for the job is received, rather than when the job is selected to run.

Cross release software

Software incompatibilities exist across releases for the VM Batch Facility software components as follows:

1. Use of any of the enhancements provided by Version 2 Release 2 necessarily requires exclusive use of Version 2 Release 2 software by any configured VM Batch Facility monitor, task machine and user interface.

2. No software prior to Version 2 Release 1 may be used in conjunction with Version 2 Release 2 software.

3. Users may use a Version 2 Release 1 user interface to submit and control jobs on a Version 2 Release 2 monitor. However, users may NOT use a Version 2 Release 2 user interface to submit and control jobs on a Version 2 Release 1 monitor.

4. All task machines owned by a Version 2 Release 2 monitor must run Version 2 Release 2 software for both the task machine and user interfaces.

5. No incompatibility exists regarding the actual jobs that may be submitted.

Installations that configure multiple VM Batch monitors must ensure that all users, monitors and task machines within the configuration conform to these requirements.

Return code changes

The return codes from all VM Batch Facility commands and monitor modules have changed. Any existing programs that interrogate these return codes for a value other than zero should be altered. The return codes are described in "VM Batch Facility Command Return Codes" on page 80.

Job spawning

If any jobs at your installation have been using unsupported techniques to issue VM Batch Facility commands (such as SUBMIT) from within the job, then these jobs may need to be modified to use the new job spawning support. Refer to the *VM Batch Facility User's Guide* for the details of job spawning.

Exploiting the enhancements provided by this release

Optional enhancements to this release that have a configuration impact are:

1. Job spawning.

   • The DGRUCD monitor exit as supplied with the product will need to be modified before a monitor will accept spawned commands.

   • A parameter is passed to monitor exits to indicate that the related job or command is a spawned job or command.

2. Storage specification by job.

   • CLASS keywords in the CONTROL FILE should include MINSTOR and MAXSTOR options to provide job submittors with a range of permissible virtual storage specifications for jobs of each class.

     Be aware that the task machines associated with the class must have been defined to CP with virtual storage values that do not conflict with the range.

- The monitor exit DGRUJB EXEC is notified of the virtual storage requirement for each job.

3. Allow jobs to be scheduled on a recurring basis without requiring users to resubmit them.

   - The DGRECUR monitor exit as supplied with the product will need to be modified to provide recurrence patterns as required by your installation.

   - A parameter is passed to monitor exits to indicate that the related job is a recurring job.

4. Save only recurring jobs to disk.

   Set the MAXIMUM-JOB-SIZE keyword of the CONTROL FILE to zero to prevent all but recurring jobs from being stored on disk by a monitor.

5. JOBSTEP notification to monitor exits.

   - The monitor exit DGRUJB EXEC is called whenever a job issues a BATCH JOBSTEP command.

   - This exit and the monitor DGRIDO exit are also notified of the last jobstep that was issued for a job.

6. The monitor exit DGRUJB EXEC is provided with a new reason 'RESTART', to distinguish it from a 'START' call.

7. All monitor exits are given an opportunity to identify themselves when they are first loaded or refreshed.

Migrating Exits

The following changes from the previous release may have an effect on exits, regardless of whether they have been customized.

1. All monitor exits must reside on the monitor's 199 disk.

2. All the monitor exits must be present on that disk. Absence of any exit constitutes an error.

3. The format of job accounting records created by the job accounting exit DGRUAC EXEC as supplied with the product has changed from fixed to variable length, and additional accounting information is provided on each record.

   If you use this exit in its unmodified form you will need to review your reporting processes that access these records.

The following changes from the previous release may have an effect on exits that have been customized.

1. All monitor exits will be called with a single parameter of '?'. You should review the structure of all your monitor exits to ensure that they will not malfunction on this call.

2. All existing monitor exits should be reviewed to ensure that they will not malfunction when provided with the additional information included in this release.

3. The DGRUJB exit should be reviewed to ensure that it will not malfunction when called with the additional reasons of 'JOBSTEP' and 'RESTART'.

4. The DGRUAC exit should be reviewed to ensure that it will not malfunction when called with the additional reason of 'RESTART'.

Incorporation of new parameters provided to the exits may be easily achieved by replacing the PARSE ARG statements in your existing exits with the corresponding PARSE ARG statements in the new sample exits. However, to make full use of the new functions offered, you should rework your existing exits based on the new samples provided. It is also recommended that your existing exits include the prologues of the new sample exits, as these serve as a guide for exit modification. Alternatively, rather than modifying your existing exits, you can extract their logic portions and incorporate them into the framework of the new samples.

If you are migrating from any release of Version 1 you should consider the following changes that were introduced with Version 2 Release 1:

1. The parameter *jobowner* in DGRUJB EXEC is changed to *jobaltuser*. Change all occurrences in your existing DGRUJB exit accordingly.

2. New parameters are passed to all exits, and the ordering of existing parameters has changed in some exits.

   In particular, a parameter is provided to some exits indicating the type of user that is the subject of the call. This is provided so that the exit may select an appropriate mechanism of communicating with the user. User types are described on page 125.

3. To complement the increased function offered by Version 2, exit DGRUJB EXEC has been split into two separate exits. The handling of abnormal events is now done in the new DGRIDO EXEC exit. The handling of normal events (RECVD, START, CHJOB and JOBEND) remains a function of DGRUJB EXEC.

   To allow immediate use of Version 2, make a copy of your existing DGRUJB EXEC, incorporating any changes as required above, named as DGRIDO EXEC.

Required RACF PERMITS

If the Resource Access Control Facility (RACF) is installed in your system, certain types of access must be granted with RACF PERMIT commands before the VM Batch Facility can be installed or used. The following table lists the types of access that must be granted.

See the RACF administrator of your system for help in meeting RACF PERMIT requirements. For more information on PERMITs and alternate user IDs, consult the Resource Access Control Facility library.

This user ID:	Must have these RACF PERMITS:
The VM Batch Facility monitor virtual machine	RACF CONTROL access to all task machine 191 minidisks.
VM Batch Facility authorized users (administrators)	RACF UPDATE access to the monitor virtual machine 199 minidisk (only for those authorized users who should have authority to customize the VM Batch Facility.)

In addition, a suitable environment must be created to allow the task machines to act as alternate user IDs for job submitters. See the *RACF Security Administrators Guide*, (SC28-1340), for information about allowing the use of alternate user IDs.

Tailoring the VM Batch Facility

When you have completed the installation and verification process described in the *VM Batch Facility Program Directory*, a basic VM Batch Facility has been created, suitable for running the verification test. See "The Basic Configuration" on page 14.

You should tailor it to meet the needs of your installation. You can do any or all of the following tasks:

Customize the VM Batch Facility monitor

Some of the additional customizing tasks that you may wish to perform are the following:

- Adding task machines to handle increased batch processing workloads and reduce the time that jobs must wait to run.

- Increasing the limits on the numbers of jobs and users that the monitor can accept at one time.

- Providing additional job classes.

A complete discussion of monitor customization is provided in , Chapter 3, "Customizing the VM Batch Facility" on page 19.

Tailor the VM Batch Facility exits

By modifying the exits, you can expand the VM Batch Facility to provide additional functions such as the following:

- Screening of jobs and commands

- Selecting default job control options

- Job accounting

- Having jobs repeat automatically

- Accepting jobs from other systems

- Having jobs be submitted from within other jobs.

For a discussion of the VM Batch Facility exits, see Chapter 5, "VM Batch Facility Installation Exits" on page 121.

Create the VMBATCH DEFAULTS file

Providing a VMBATCH DEFAULTS file can give increased convenience to users of the VM Batch Facility. See Chapter 6, "VMBATCH DEFAULTS File" on page 159 for more information.

Have the VM Batch Facility started automatically

To autolog the VM Batch Facility whenever you IPL your system you should add a CP AUTOLOG command to the PROFILE EXEC of the AUTOLOG1 user ID (or whichever user ID you use to automatically log on other virtual machines).

Multiple Facilities.

You are not restricted to running a single VM Batch Facility. You can distribute your job load among multiple monitors if required. Furthermore, if you have RSCS links to other systems running VM Batch Facility monitors, jobs may be directed to them. For a discussion on configuring and using multiple and remote VM Batch Facilities, see Chapter 7, "Multiple VM Batch Facilities" on page 165.

The Basic Configuration

The basic VM Batch Facility is set up with these characteristics:

- It consists of three virtual machines: a monitor virtual machine, in which the VM Batch Facility control program runs; and two task machines, available for running jobs. Both task machines are defined to be available for running jobs 24 hours a day. Both machines can accept job class A and job class B.

- Limits are 50 users and 500 jobs at any one time.

- Limits of 8 running and 32 queued or running jobs per user are allowed at any one time. The VM Batch Facility will reject new jobs from any user attempting to exceed these limits.

- No job accounting is provided.

- Load Level Scheduling is not active.

- All installation exits provide default function.

- The basic VM Batch Facility has three job classes available: A, B, and AUTOLOG.

Job class A is defined with a 24-hour a day start time window. That means that jobs submitted without start window times can start at any time of the day. It is intended for small and medium-sized jobs. The job class is defined with the following limits:

  – Default limit on processor time used by jobs: 1000 seconds.

  – Maximum processor time permitted to jobs: 2000 seconds.

  – Default limit on virtual printer output by jobs: 64,000 records.

  – Maximum permitted output of virtual printer records: 1,000,000 records.

  – Default limit on virtual punch output by jobs: 64,000 records.

  – Maximum permitted output of virtual punch records: 1,000,000 records.

  – Virtual storage to be available in task machines: 2048K (2MB). Both task machines will be autologged with this amount of virtual storage when they run class A jobs.

  – Jobs will not be subject to LLS monitoring.

  – Stall detection and reporting thresholds will default to the VM Batch Facility global values.

Job class B is defined with a night-time start time window of 1700 to 0800 (5 p.m. to 8 a.m.). Class B jobs whose owners did not specify another start time window will be started only between 5 p.m. and 8 a.m. The job class is defined with these limits:

- Default limit on processor time used by jobs: 9999 seconds.

- Maximum processor time permitted to jobs: unrestricted.

- Default limit on virtual printer output by jobs: 64,000 records.

- Maximum permitted output of virtual printer records: 1,000,000 records.

- Default limit on virtual punch output by jobs: 64,000 records.

- Maximum permitted output of virtual punch records: 1,000,000 records.

- Virtual storage to be available in task machines: 4096K (4MB) will be provided by default when the task machines are autologged to run class B jobs. However, job submittors may request storage as high as 16M or as low as 2M for their class B jobs.

- Stall detection and reporting will be suppressed.

- Jobs will not be subject to LLS monitoring by default.

The AUTOLOG job class lets your users submit jobs and have them run on their own user IDs.

How the Task Virtual Machines are Defined

The basic VM Batch Facility provides two task virtual machines. You can define up to 128 task machines.

Use the following sample directory entry to define additional task machines. Note that task virtual machines must not have secondary consoles defined in their CP directory entries.

```
┌─── Sample Task Machine Directory Entry ───────────────────────┐
│ USER taskid password 2M 16M G                                 │
│ ACCOUNT acctid distcode                                       │
│ IPL CMS                                                       │
│ CONSOLE 01F 3215                                              │
│ SPOOL 00C 2540 READER *                                       │
│ SPOOL 00D 2540 PUNCH A                                        │
│ SPOOL 00E 1403 A                                              │
│ LINK MAINT 190 190 RR                                         │
│ LINK MAINT 19E 19E RR                                         │
│ * Work disk                                                   │
│ MDISK 191 3380 nnn 004 volser MR pr pw pm                     │
└───────────────────────────────────────────────────────────────┘
```

Notes:

USER *taskid password*
   Be sure to match the user IDs and passwords of your task machines with their entries in the CONTROL FILE.

2MB 16MB
   Be sure to match the storage definitions of your task machines with their associated CLASS ranges stated in the CONTROL FILE.

G

   Specify user class G unless a task machine will have to run execs that require a higher user class authorization.

IPL CMS

An IPL of CMS in a task machine is mandatory.

SPOOL 00C 2540 READER *

Task machines must have a virtual reader defined at virtual address 00C.

SPOOL 00D 2540 PUNCH A

Task machines must have a virtual punch defined at virtual address 00D.

SPOOL 00E 1403 A

Task machines must have a virtual printer defined at virtual address 00E.

LINK MAINT 19E 19E RR

This LINK statement is required; VM Batch Facility user interface files are stored on the MAINT 19E minidisk.

You may also wish to add additional link statements for minidisks that will be routinely needed by jobs.

MDISK

A 191 minidisk is mandatory for each task machine.

Note: On heavily loaded systems it is recommended that the CP PRIORITY or SHARE allocated to task virtual machines be set such, that the execution of jobs by these machines does not adversely affect response times for interactive users.

How the Monitor Virtual Machine is Defined

```
┌─ Sample Monitor Machine Directory Entry ──────────────────────────┐
│ USER BATCH password 2M 4M ABEG                                     │
│ ACCOUNT acctid distcode                                           │
│ OPTION MAXCONN 256                                                │
│ IUCV ALLOW                                                        │
│ IPL CMS                                                           │
│ CONSOLE 01F 3215                                                  │
│ SPOOL 00C 2540 READER *                                          │
│ SPOOL 00D 2540 PUNCH A                                           │
│ SPOOL 00E 1403 A                                                │
│ LINK MAINT 190 190 RR                                            │
│ LINK MAINT 19E 19E RR                                            │
│ * VM Batch Facility executable material:                        │
│ LINK P684137A 196 192 RR                                        │
│ * VM Batch Facility PROFILE EXEC:                               │
│ MDISK 191 3380 nnn 001 volser MR pr pw pm                      │
│ * Job storage:                                                  │
│ MDISK 193 3380 nnn 010 volser MR pr pw pm                      │
│ * Monitor work files (control blocks)                          │
│ MDISK 194 3380 nnn 002 volser MR pr pw pm                      │
│ * Accounting data                                               │
│ MDISK 195 3380 nnn 001 volser MR pr pw pm                      │
│ * VM Batch Facility control file and monitor exits:            │
│ MDISK 199 3380 nnn 001 volser MR pr pw pm                      │
└───────────────────────────────────────────────────────────────────┘
```

Notes:

USER BATCH

This is the default userid for the VM Batch Facility monitor virtual machine. Change it if required. If changed then users will need to issue the BATCH

ADDRESS command to direct their jobs to the monitor, or a default address may be provided in the VMBATCH DEFAULTS File.

ABEG

In all VM systems, the monitor machine must have the standard CP A, B, E, and G class privileges.

The following commands and diagnose codes, issued by the monitor machine, require greater than class G privilege:

- WARNING
- FORCE
- AUTOLOG
- DIAGNOSE X'04'
- DIAGNOSE X'84'
- DIAGNOSE X'D4' subcode X'0000' (non-RACF B1 systems)
- DIAGNOSE X'D4' subcode X'0004' (RACF B1 systems only)
- MSGNOH
- INDICATE USER
- INDICATE QUEUES
- LOCATE
- XAUTOLOG

OPTION MAXCONN 256, IUCV ALLOW

The MAXCONN 256 option and the IUCV ALLOW statement are required.

SPOOL 00C 2540 READER *

Monitor machines must have a virtual reader defined at virtual address 00C.

SPOOL 00D 2540 PUNCH A

Monitor machines must have a virtual punch defined at virtual address 00D.

LINK P684137A 196 192 RR

This LINK statement (or equivalent) is required; VM Batch Facility monitor installation exits and the CONTROL FILE are stored on the 199 minidisk.

MDISK

Monitor machine minidisks must be defined at the addresses shown in the MDISK statements. Passwords are optional if the Resource Access Control Facility (RACF) is installed in your system. Be sure to size these according to your requirements.

Installing the VM Batch Facility

Chapter 3. Customizing the VM Batch Facility

You can tailor many of the operating characteristics of the VM Batch Facility to meet your installation's needs. Some of the customizing tasks you may wish to perform are listed below.

- Reviewing the supplied CONTROL FILE. See page 20.

- Reviewing current settings of keywords in the CONTROL FILE. See page 22.

- Reviewing the customization process: Editing the CONTROL FILE and putting your changes into effect. See page 23.

- Tailoring the VM Batch Facility, changing its capacity for jobs. See page 25.

    - Reviewing sample configurations of task machines and job classes, and suggestions for coordinating job classes and task machines. See page 26.

    - Adding or deleting task machines. See page 31.

    - Adding or deleting job classes. See page 34.

    - Raising or lowering maximum user and job limits. See page 41.

    - Tailoring the VM Batch Facility monitor machine, if required to handle an increased batch processing work load. See page 45.

- Identifying a user as a VM Batch Facility administrator or authorized user. See page 49.

- Specifying the default alert message receiver. See page 49.

- Adjusting the length of time that jobs can remain stalled. See page 51.

- Adjusting the number of jobs that job owners can have queued and running. See pages 54 and 55.

- Specifying the maximum size of job files that can be stored on the monitor machine 193 minidisk. See page 56.

- Specifying the type of VM Batch Facility accounting support. See page 57.

- Specifying the default state of console message spooling. See page 59.

- Designating the default console message receiver. See page 59.

- Adjusting the dispatch interval (the longest time that the VM Batch Facility will wait before starting jobs that are ready to run.) See page 61.

- Adjusting the resource checking interval (the longest time that the VM Batch Facility will wait before checking on resource usage of running jobs). See page 62.

- Adjusting the time allowed for large jobs to initialize. See page 63.

- Tuning of the Load Level Scheduling (LLS) activity. See page 64.

Predefined Keyword Settings

The VM Batch Facility is shipped to you with a predefined CONTROL FILE that has values already set for most of the customizing keywords. The predefined CONTROL FILE is reprinted below.

```
 *---------------------------------------------------------------------*
 *  VM BATCH FACILITY   5684-137                                       *
 *  (C) COPYRIGHT IBM CORPORATION 1990-1993                            *
 *  LICENSED MATERIAL - PROGRAM PROPERTY OF IBM                        *
 *  REFER TO COPYRIGHT INSTRUCTIONS: FORM G120-2083                    *
 *                                                                     *
 *                   VM BATCH FACILITY CONTROL FILE                    *
 *---------------------------------------------------------------------*


 *---------------------------------------------------------------------*
 *                         AUTHORIZED USERS                            *
 *---------------------------------------------------------------------*
  ADMIN XXXXXXXX
  ADMIN XXXXXXXX XXXXXXXX
 *---------------------------------------------------------------------*
 *                          TASK MACHINES                             *
 *---------------------------------------------------------------------*
 TASK BATCH1  XXXXXXXX XXXXXXXX  .AB  0000 2400
 TASK BATCH2  XXXXXXXX XXXXXXXX  .AB  0000 2400
```

Figure 1 (Part 1 of 2). Default VM Batch Facility CONTROL FILE

```
*----------------------------------------------------------------------*
*                          CLASSES                                     *
*----------------------------------------------------------------------*
*
* This sample Class A :
*    - does not allow a storage override
*    - does not subject jobs to LLS
*    - stall values default
 CLASS CLASSID=A DEFCPU=1000 MAXCPU=2000 +
       DEFPRT=64 MAXPRT=1000 +
       DEFPUN=64 MAXPUN=1000 +
       DEFSTOR=2M +
       LLSDEF=LLS
*
* This sample Class B :
*    - allows unlimited CPU if requested
*    - allows storage values between 2M and 16M
*    - stall processing is suppressed
*    - has a restricted window
*    - does not subject jobs to LLS by default.
 CLASS CLASSID = B DEFCPU = 9999 MAXCPU = NONE +
       DEFPRT=64 MAXPRT=1000 +
       DEFPUN=64,MAXPUN=1000 +
       DEFSTOR=4M MINSTOR=2M MAXSTOR=16M +
       STALLWRN=* +
       SWINDOW=1700 EWINDOW=0800
*
 CLASS CLASSID=AUTOLOG


*----------------------------------------------------------------------*
*                          LLS ACTIVITY                                *
*----------------------------------------------------------------------*
*  LLS RESOURCE=CPU SCV=XXXX RCD=XXXX


*----------------------------------------------------------------------*
*                       CONTROL PARAMETERS                             *
*----------------------------------------------------------------------*
*
MAXIMUM-USERS          50
MAXIMUM-JOBS          500
*
DISPATCH-INTERVAL      4
CHECK-RUNNING-JOBS    10
MAXJOBS/USER:RUNNING   8
MAXJOBS/USER:RUNNING+QUEUED  32
MINIMUM-STALL-DELAY    60
MAXIMUM-STALL-WARNINGS 3
MAXIMUM-JOB-SIZE  100000
JOB-INIT-TIME/100000-RECS  4
ACCOUNT 0
CONSOLE BATCH
*
```

Figure 1 (Part 2 of 2). Default VM Batch Facility CONTROL FILE

Reviewing Current CONTROL FILE Settings

The box below lists the three commands needed to review the CONTROL FILE entries currently in effect in the VM Batch Facility.

Reviewing Current Customizing Options

To review all keywords currently in effect except for CLASS and TASK, enter this command:

```
batch control
```

To review job class definitions (CLASS keyword entries), enter one of these commands:

`BATCH Query Class ALL`	To review all job class definitions.
`BATCH Query Class classid`	To review the definition of one class.

To review task machine definitions (TASK keyword entries), enter one of these commands:

`BATCH Query Task ALL`	To review the definitions and status of all task machines.
`BATCH Query Task taskid`	To review the definition and status of one task machine.

Notes

1. **Messages:** All three commands return keyword values to you as messages displayed at your terminal.

2. **Passwords not displayed:** The BATCH QUERY TASK command does not display the passwords in TASK entries. To review these passwords, edit the CONTROL FILE.

3. **Keywords in effect may differ from keyword settings in the CONTROL FILE:** The keywords and values shown in the BATCH CONTROL messages reflect the copy of the CONTROL FILE that is actually in effect in the VM Batch Facility. If you have changed the CONTROL FILE but have not put the file into effect yet, the values in the messages will differ from the values in the file. Also, if there are errors in the CONTROL FILE, the keywords with invalid values may be set to their predefined values. See page 20 for the predefined values.

Editing the CONTROL FILE and Putting Changes into Effect

The following box lists the steps you must perform to change the VM Batch Facility customizing options and put the changes into effect.

Customizing the VM Batch Facility

1. Edit the CONTROL FILE.

   Refer to the *Program Directory* for detailed instructions on how to update the CONTROL FILE.

2. Enter this command to put the altered CONTROL FILE into effect:

   ```
   batch control set
   ```

   You can also put a new CONTROL FILE into effect by stopping the VM Batch Facility and restarting it. The VM Batch Facility automatically reprocesses the CONTROL FILE every time it is started.

   Note: If you are changing the MAXIMUM-JOBS or MAXIMUM-USERS keywords, you must stop and restart the VM Batch Facility. The VM Batch Facility will reject changes to these keywords if you enter BATCH CONTROL SET to put the changes into effect. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for more information.

Editing the Control File

When you edit the file, remember the following:

- Each Control File statement must start on a new line.
- You can enter up to 72 characters per line.
- Comment lines, which will not be processed, start with an asterisk (*).
- No commas are allowed in numeric values: enter ten thousand as 10000, not as 10,000.
- There are two types of statements in the Control File

  - Some statements have their parameters as values only that are in fixed positions, such as the TASK and ADMIN statements.

    For these statements, the entire definition must be on one line, in the specified order. No continuation is permitted.

  - Other statements specify their parameters as keyword=value pairs, such as the CLASS and LLS statements.

    For these statements:

    - An = is required between a keyword and its value.

    - keyword=value pairs may be specified in any order.

    - A statement may occupy more than one line. + is the continuation character.

    - Both comma and blank are valid delimiters.

Notes

1. **Dynamic customizing:** You can make changes to the CONTROL FILE and put them into effect without stopping the VM Batch Facility. The VM Batch Facility uses a copy of the CONTROL FILE while it is running, so editing the CONTROL FILE has no effect on the VM Batch Facility.

2. **Errors in the CONTROL FILE:** If required keywords are missing or have invalid values, the VM Batch Facility does the following:

   - When the VM Batch Facility is started, the incorrect or missing keywords will be set to their default values. These values are shown on page 20. You will receive messages identifying the errors found in the CONTROL FILE.

     Exceptions: The VM Batch Facility will stop running if there are no valid TASK, ADMIN, or CLASS keywords in the CONTROL FILE.

   - When you have entered a BATCH CONTROL SET command, the VM Batch Facility will continue to operate using the predefined values of keywords, where available. The predefined values are shown on page 20. You will receive messages identifying the errors found in the CONTROL FILE.

     Exceptions:

       - The VM Batch Facility will not have any task machines if the new CONTROL FILE did not have any valid TASK entries.
       - The VM Batch Facility will use only job class A, or job class AUTOLOG, if there were no valid CLASS entries in the new file. Job class AUTOLOG is used only if CLASS AUTOLOG was the only job class defined to the VM Batch Facility.
       - If there are no valid ADMIN entries, only the user who entered the BATCH CONTROL SET will be recognized as an authorized user.

   If there is an error in the CONTROL FILE, be sure to correct it before stopping and restarting the VM Batch Facility. The VM Batch Facility may not be able to restart if there are errors in the CONTROL FILE.

Tailoring the Job and User Capacity of the VM Batch Facility

This section explains how to tailor the size and capacity of the VM Batch Facility to meet the batch processing work load in your system.

To tailor the capacity of the VM Batch Facility, do one or more of these customizing tasks:

- Add or delete task machines.

- Add or delete job classes.

- Change the limit on the number of jobs that the VM Batch Facility can accept at one time.

- Change the limit on the number of users who can have jobs in progress at one time in the VM Batch Facility.

Changing the limits on jobs and users that the VM Batch Facility can accept may also require you to tailor the monitor virtual machine. Instructions for doing that are included in this section.

The following table lists some of the reasons for tailoring the VM Batch Facility.

You may wish to make this change:	For these reasons:
Delete task machines.	1. Reduce the amount of system resources used by the VM Batch Facility. Using only a single task machine conserves resources in a small system. You may also wish to lower MAXIMUM-USERS and MAXIMUM-JOBS limits to levels appropriate for a small system. 2. Have the VM Batch Facility accept only CLASS AUTOLOG jobs. Task machines are not required if the only job class you define is the AUTOLOG class.
Add task machines.	1. Shorten the time that jobs must wait before being run. (By assigning multiple task machines to the same job class, you can shorten the turn around time for jobs in that class.) 2. Handle a larger work load than the two default task machines can support. If your system has many users or has a heavy batch processing work load, you may wish to add both task machines and job classes. You may also wish to raise the limits on the numbers of jobs and users that the VM Batch Facility can accept.
Raise or lower the limit on the number of jobs that the VM Batch Facility can accept at one time.	Adjust the maximum jobs limit in accordance with the batch processing work load in your system and the resources available to the VM Batch Facility. Raising the limit substantially may require you to tailor the VM Batch Facility monitor machine to ensure that the VM Batch Facility works properly. Instructions for tailoring the monitor machine are shown in "Tailoring the Monitor Machine to Handle Higher Work Loads" on page 45. You can also control the number of jobs each user may have in progress or queued at one time with the MAXJOBS/USER:RUNNING+QUEUED and the MAXJOBS/USER:RUNNING keywords. See pages 54 and 55.
Raise or lower the limit on the number of users that the VM Batch Facility can accept at one time.	Allow more or fewer users to have jobs queued and/or running at the same time. If your system has many users, you may wish to raise the default limit on the number of users that the VM Batch Facility can accept. If your system has only a small number of users, you may wish to lower the limit from the default value of 50 users.

Sample Configurations

The following sample configurations show how machines, classes, job and user limits are put together.

```
┌─── Separating Large and Small Jobs ──────────────────────────────────┐
│                                                                       │
│ ***  CONTROL FILE  ***                                                │
│ * Task machines LARGE1 and LARGE2 accept CLASS LARGE jobs             │
│ * and run only at night                                               │
│ TASK LARGE1 logon_cpssword write_cpssword LARGE 1700 0500             │
│ TASK LARGE2 logon_cpssword write_cpssword LARGE 1700 0500             │
│ *                                                                     │
│ * Task machines SMALL1 and SMALL2 accept CLASS A jobs                 │
│ * and run 24 hours a day                                              │
│ TASK SMALL1 logon_cpssword write_cpssword .A 0000 2400                │
│ TASK SMALL2 logon_cpssword write_cpssword .A 0000 2400                │
│ *                                                                     │
│ * Job CLASS LARGE runs large jobs                                     │
│ CLASS CLASSID=LARGE SWINDOW=1700 EWINDOW=0500 +                       │
│       DEFCPU=9999 MAXCPU=9999 DEFPRT=9999 MAXPRT=9999  +              │
│       DEFPUN=9999 MAXPUN=9999                                         │
│ *                                                                     │
│ * Job Class A is the default class and is intended for small jobs     │
│ CLASS CLASSID=A SWINDOW=0000 EWINDOW=2400 +                           │
│       DEFCPU=600 MAXCPU=600 DEFPRT=20 MAXPRT=20 +                     │
│       DEFPUN=20 MAXPUN=20                                             │
│ *                                                                     │
│ * Maximum users and jobs limits                                       │
│ MAXIMUM-USERS 200                                                     │
│ MAXIMUM-JOBS 1000                                                     │
│                                                                       │
└───────────────────────────────────────────────────────────────────────┘
```

In this configuration:

- Big jobs are run only at night on task machines LARGE1 and LARGE2. These task machines accept jobs only from job class LARGE. Only job class LARGE has resource limits that are high enough to allow big jobs to run to completion.

  By defining night-time start windows on task machines LARGE1 and LARGE2, the configuration assures that big jobs will run only at night. Jobs can be run only when there is a task machine available to accept them.

- CLASS LARGE is defined with the same start time window as the LARGE1 and LARGE2 task machines. It is expected that most jobs will be submitted to CLASS LARGE without a start time window. In that case, the job class window is used, so jobs will automatically have start time windows that match the task machine windows.

  A job submitted with a user specified start time window that is not compatible with the task machine configuration is not rejected. The job is held until the task machine is reconfigured or the start window is changed with the CHJOB command.

- The task machines with the SMALL1 and SMALL2 user IDs accept jobs only from CLASS A, the default job class. CLASS A is set up to run jobs 24 hours a day, but has resource limits that will allow only small jobs to run to completion.

```
┌─  Accommodating a Heavy Batch Processing Work Load  ─────────┐
│                                                              │
│ * NIGHT1, NIGHT2, and NIGHT3 accept CLASS NIGHT jobs         │
│ * and run only at night                                      │
│ TASK NIGHT1 logon_cpssword write_cpssword NIGHT 1800 0700    │
│ TASK NIGHT2 logon_cpssword write_cpssword NIGHT 1800 0700    │
│ TASK NIGHT3 logon_cpssword write_cpssword NIGHT 1800 0700    │
│ *                                                            │
│ * A1 through A8 accept small and medium-sized CLASS A jobs   │
│ * and run 24 hours a day                                     │
│ TASK A1 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A2 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A3 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A4 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A5 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A6 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A7 logon_cpssword write_cpssword .A 0000 2400           │
│ TASK A8 logon_cpssword write_cpssword .A 0000 2400           │
│ *                                                            │
│ * SPECIAL accepts jobs from CLASS SPECIAL, jobs that run     │
│ * with extra virtual storage                                 │
│ TASK SPECIAL logon_cpssword write_cpssword SPECIAL 1800 0700 │
│ *                                                            │
│ * Job Class NIGHT runs large jobs at night.                  │
│ CLASS CLASSID=NIGHT SWINDOW=1800 EWINDOW=0700 +              │
│       DEFCPU=9999 MAXCPU=9999 DEFPRT=9999 MAXPRT=9999 +      │
│       DEFPUN=9999 MAXPUN=9999                                │
│ *                                                            │
│ * Job Class A is the default class; for small and medium-sized jobs │
│ CLASS CLASSID=A SWINDOW=0000 EWINDOW=2400 +                  │
│       DEFCPU=1000 MAXCPU=1000 DEFPRT=40 MAXPRT=40 +          │
│       DEFPUN=40 MAXPUN=40                                    │
│ *                                                            │
│ * Job Class SPECIAL runs jobs that require extra virtual storage │
│ CLASS CLASSID=SPECIAL SWINDOW=0000 EWINDOW=2400 +            │
│       DEFSTOR=12M MINSTOR=8M MAXSTOR=16M +                   │
│       DEFCPU=9999 MAXCPU=9999 DEFPRT=9999 MAXPRT=9999 +      │
│       DEFPUN=9999 MAXPUN=9999                                │
│ *                                                            │
│ * Maximum users and jobs limits                              │
│ MAXIMUM-USERS 500                                            │
│ MAXIMUM-JOBS 4000                                            │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

In this configuration:

- The MAXIMUM-JOBS and MAXIMUM-USERS limits have been raised to accommodate a larger population of users.

- CLASS A serves as the job class for the majority of jobs. Most users are expected to submit their jobs without a job class, so most jobs will automatically be assigned to job class A.

  The resource limits of job class A permit small and medium size jobs to run. But the resource limits would cause large jobs to be canceled before being completed.

- Eight task machines have been defined to accept CLASS A jobs. These task machines have start time windows that make them available 24 hours a day. If a backlog of CLASS A jobs develops during the business day, the task machines can run through the night to finish the jobs.

- Two other job classes have been defined, together with extra task machines.

   The NIGHT job class lets users run large and small jobs during off-peak hours, from 6 p.m. to 7 a.m.

   The NIGHT job class and the NIGHT task machines have been defined with identical start time windows. It is expected that most jobs will be submitted to CLASS NIGHT without start time windows. The job class start time window will be used for those jobs, so that job and task machine start windows will match.

   A job submitted with a user specified start time window that is not compatible with the task machine configuration is not rejected. The job is held until the task machine is reconfigured or the start window is changed with the CHJOB command.

   The SPECIAL job class is intended for a small number of jobs that require a large amount of virtual storage in which to run. To ensure that SPECIAL jobs can run, the CP directory entry of the SPECIAL task machine would have to allow the task machine to have up to 16MB of virtual storage.

Suggestions for Coordinating Job Classes and Task Machines

Here are some suggestions for ensuring that the task machines and job classes in your configuration will work properly together:

- When you define multiple job classes, be sure to coordinate the job classes with the task machines available to the VM Batch Facility:

    - Make sure that every job class is accepted by at least one task machine. Jobs are put in HOLD status if there is not a task machine that can accept the job class to which they belong. Jobs in HOLD status will not be started.

    - Define several task machines for a class that has a high volume of jobs or urgent jobs. Having several task machines available ensures that jobs will be started more quickly and lessens the chance that a backlog of queued jobs will develop.

        Other classes can share a task machine allocated to an important job class. See "Filling Out the TASK Keyword" on page 31 for more information.

    - Split one task machine among several job classes that are not used frequently. Each task machine can accept jobs from up to seven job classes.

- Be sure that the CP directory entry of a task machine allows enough virtual storage to accommodate any job class that the machine can accept. When it logs on a task machine, the VM Batch Facility requests the amount of virtual storage defined for the job. The CLASS entry defines the range of virtual storage permissible for its jobs. If the CP directory entry does not allow enough virtual storage, the job will not be started. The VM Batch Facility will mark the task machine as OFFLINE and will stop using it.

- If it is important that jobs start only during the start window of a job class, make sure that the task machines that accept the job class have the same start window as the job class.

The start time window of a job class is used only as a default–it applies to a job only if the job owner does not specify a start time window. If the job owner specifies a start time window, it overrides the job class window if there is any conflict between them.

However, the task machine start time window can enforce the job class window. Jobs can start only when a task machine is available, no matter what start time window is specified by the job owner. When a job is submitted with a user specified start window that is not compatible with any task machine, the job is placed in HOLD status until either the job or task machine start window is changed.

Here is an example to show the relationship between task machine, job class, and user-specified start windows. The sample job class and task machine below have the same start time window: 2000 (8 p.m.) to 0500 (5 a.m.):

```
CLASS CLASSID=NIGHT SWINDOW=2000 EWINDOW=0500 +
      DEFCPU=600 MAXCPU=600 DEFPRT=25 MAXPRT=25 +
      DEFPUN=25 MAXPUN=25

TASK NIGHT cpssword write_cpssword  NIGHT 2000 0500
```

User CHARLIE submits a job to job CLASS NIGHT, and specifies a different start time window: 1800 to 0730. CHARLIE's start window times override the job class times, so his job is eligible to start as early as 1800 (6 p.m.). But the VM Batch Facility will not start CHARLIE's job until the NIGHT task machine becomes available at 2000.

Adding and Deleting Task Machines

┌─── **Adding and Deleting Task Virtual Machines: Required Tasks** ───┐

1. To add task machines, start by adding CP directory entries for the new task machines. See "How the Task Virtual Machines are Defined" on page 15 for sample directory entries.

2. Obtain WRITE access to the task virtual machine's 191 minidisk, and format it ready for use.

3. Edit the CONTROL FILE.

   Refer to the *Program Directory* for detailed instructions on how to update the CONTROL FILE.

   Add task machines by adding TASK keyword entries to the CONTROL FILE, to identify the task machines to the VM Batch Facility. Information on filling out the TASK keyword starts at "Filling out the TASK Keyword," on the next page.

   Delete task machines by deleting the TASK entries that define the machines. **Caution:** Before you delete a task machine, be sure that all job classes still have task machines that can accept them. If you delete the only task machine that can accept a job class, all jobs in that class will be put in HOLD status. The VM Batch Facility will not be able to start the jobs. Also, task machines running jobs should not be deleted.

4. If the Resource Access Control Facility (RACF) is installed in your system, enter the RACF PERMITs required for the monitor machine to use new task machines for running jobs. See "Required RACF PERMITS" on page 12 for a list of the required PERMITs.

5. Enter the BATCH CONTROL SET command or stop and restart the VM Batch Facility to put the changes you have made into effect.

└──┘

Filling Out the TASK Keyword

Use the TASK keyword to identify a task virtual machine to the VM Batch Facility control program. The VM Batch Facility can use a task machine to run jobs only if it is identified with a TASK keyword entry in the CONTROL FILE.

```
┌─ Format-1 ─────────────────────────────────────────────────────────┐
│                                                                     │
│  ►►──TASK──userid──password1──┬─password2─┬──classids──tttt──tttt──► │
│                               └─*NOPASS*──┘                         │
│                                                                     │
│  ►──secure──►◄                                                      │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

```
┌─ Format-2 ─────────────────────────────────────────────────────────┐
│                                                                     │
│  ►►──TASK──userid──*NOPASS*──┬─password2─┬──classids──tttt──tttt──►  │
│                              └─*NOPASS*──┘                          │
│                                                                     │
│  ►──secure──password3──►◄                                           │
│                                                                     │
└─────────────────────────────────────────────────────────────────────┘
```

Use Format 2 when you have enabled password suppression on your system, so that passwords will not be used when task machines are AUTOLOGed.

userid

Enter the user ID of the task virtual machine.

password1

Enter the logon password of the task virtual machine. The logon password must be the first password listed in the TASK keyword, and is used for autologging the task machine.

If logon passwords are suppressed on your system, that is, passwords are not required for autologging, then enter '*NOPASS*' for *password1*. If you enter *NOPASS* for *password1*, then the parameter *password3* on page 33 is mandatory. Also, you may not configure the VM Batch Facility to insist on passwords, when logon passwords are suppressed. Refer to Chapter 6, "VMBATCH DEFAULTS File" on page 159 for details.

password2

Enter the WRITE password of the task machine 191 minidisk. This password must be the second password listed in the TASK keyword.

If your system does not require link passwords, enter *NOPASS*.

classid(s)

Enter the names of the job classes that this task machine can accept. The class IDs must be either type 1 or type 2; a task machine cannot accept both types of job classes. (Type 1 job classes have a single letter or number as the class ID. Type 2 job classes have a class ID that is from 2 to 8 characters long.)

If the task machine is to accept only **type 1 job classes:**

- You can enter from one to a maximum of seven job classes.

- Enter the class IDs in capital letters.

- The total list of classes cannot be more than 8 characters long, and has to start with a period, like this:

    `.ABCDEFG`

- The order in which you enter the job classes determines their priority on the task machine. The first job class has the highest priority; the last has the lowest.

  The VM Batch Facility uses the priority of classes in selecting the next job to run on a task machine. A job from a high priority class will be started before a job from a lower priority class, within any start time or other limitations.

If the task machine is to accept only **type 2 job classes:**

- Enter up to 8 characters to identify the job classes that can be accepted on the task machine. The task machine can accept jobs from any classes whose class IDs are subsets of the characters you have entered, starting from the left. The characters that you enter do not have to identify a valid class ID, read in total.

  Example: If you enter XYZAB, the task machine will accept jobs from these job classes, if they have been defined in the CONTROL FILE:

  - XY
  - XYZ
  - XYZA
  - XYZAB

- The length of each type 2 job class accepted by a task machine determines the priority of the job class on the task machine. Longer class IDs have higher priority.

  For example, a job assigned to class XYZAB has higher priority than a job assigned to class XYZ. Class XYZ, in turn, has higher priority than job class XY.

tttt

Enter the time of day when the VM Batch Facility can start using this task machine to run jobs. This time is the start time of the *start time window*. The start time window defines the period of time each day during which the VM Batch Facility can start jobs on the task machine. (Note that once a job starts to run, it keeps running, even if the start window of the task machine closes.)

Enter the start time in 24-hour time notation. Do not use a colon or any other nonnumeric character in the time. For example, enter 1700 for 5 p.m., 2359 for 11:59 p.m.

To make a task machine available 24 hours a day, enter 0000 for the start time and 2400 for the end time of the start time window.

tttt

Enter the time of day when the VM Batch Facility is no longer allowed to start jobs in this task machine. (Note that a running job will continue running after this time arrives. The value limits the ability of the VM Batch Facility to start jobs; it does not affect running jobs.)

This time is the end time of the *start time window*. The start time window defines the period of time each day during which the VM Batch Facility can start jobs on the task machine.

Enter the end time in 24-hour time notation. Do not use a colon or any other nonnumeric character in the time. For example, enter 1700 for 5 p.m., 2359 for 11:59 p.m.

secure
> Enter Y to cause the task machine to overwrite residual data on its 191 minidisk and to erase all files from the directory. Enter N to erase files from the directory only.

password3
> Enter the VM Directory User ID password if you entered *NOPASS* for *password1* on the TASK statement. The password is required for DIAG X'84', even though it is not used for logging on.

Usage Notes

1. **CLASS AUTOLOG:** You should not define TASK keywords for virtual machines used to run CLASS AUTOLOG jobs. (CLASS AUTOLOG jobs run on job owners' virtual machines.)

2. **Coordinating task machines and job classes:** See "Sample Configurations" on page 26 for suggestions on assigning job classes to your new task machines.

3. **Changing task machine start windows or job classes:** Changing the start time window of a task machine or the job classes that it can accept can cause queued jobs to be put in HOLD status.

   This problem can occur if, after the change, there is no task machine that has a start window that overlaps with the jobs' start windows and/or no task machine that can accept the job classes of the jobs. The jobs are held until the task machine is reconfigured or the start window is changed with the CHJOB command.

4. **Maximum number of task machines:** You may define a maximum of 128 task machines.

Adding and Deleting Job Classes

Adding and Deleting Job Classes: Required Tasks

To add or delete one or more job classes, follow these instructions:

1. Edit the CONTROL FILE.

   Refer to the *Program Directory* for detailed instructions on how to update the CONTROL FILE.

   Add CLASS keyword entries to define job classes, or erase CLASS entries to delete job classes. Information on filling out the CLASS keyword starts at "Filling Out the CLASS Keyword," on the next page.

   Caution: Do not delete job class A unless only job class AUTOLOG is to be defined in the CONTROL FILE.

2. Enter the BATCH CONTROL SET command or stop and restart the VM Batch Facility to put the changes you have made into effect.

Filling Out the CLASS Keyword

Use the CLASS keyword to define a job class to the VM Batch Facility.

A job class specifies default and maximum job control options (start window times, and resource usage limits that apply to jobs run under that class. It also specifies the amount of virtual storage that will be available in a task machine that is logged on to run a job belonging to the class. Other optional parameters control the system's stall activity and job subjection to LLS.

Effectively, a job class identifies a type of job that can be accepted by the VM Batch Facility. Defining job classes enables you to divide jobs into groups that will run with different resource limits and default start time windows.

The VM Batch Facility assigns jobs to the classes specified by job owners, or to job class A if a job owner does not specify a class.

Before changing job classes, you may wish to review "Sample Configurations" on page 26. This section gives suggestions for organizing multiple job classes.

```
┌─ Format ────────────────────────────────────────────────────────────┐
│                              ┌─SWINDOW=0000─┐   ┌─EWINDOW=2400─┐      │
│  ►►──CLASS──CLASSID=classid──┴─SWINDOW=tttt─┘   └─EWINDOW=tttt─┘──►   │
│                                                                      │
│                                    ┌─nnnnnnnn─┐                       │
│  ►──DEFCPU=nnnnnnnn──MAXCPU=───────┴─NONE─────┴──DEFPRT=nnnn──────►   │
│                                                                      │
│         ┌─nnnn─┐                         ┌─nnn──┐                     │
│  ►──MAXPRT=─┴─NONE─┴──DEFPUN=nnnn──MAXPUN=──┴─NONE─┴──────────────►   │
│                                                                      │
│       ┌─DEFSTOR=*────────────┐        ┌─MINSTOR=*────────────┐       │
│  ►────┴─DEFSTOR──=──┬─nnnnnnnnK─┬─┘    └─MINSTOR=──┬─nnnnnnnnK─┬─┘──► │
│                     └─nnnnM─────┘                  └─nnnnM─────┘      │
│                                                                      │
│       ┌─MAXSTOR=*────────────┐                                       │
│  ►────┴─MAXSTOR=──┬─nnnnnnnnK─┬─┘    └─STALLDLY=tttt─┘────────────►   │
│                   └─nnnnM─────┘                                       │
│                                    ┌─LLSDEF=NOLLS─┐  ┌─LLSOVER=YES─┐  │
│  ►────┴─STALLWRN=nnnn─┘────────────┴─LLSDEF=LLS───┘  └─LLSOVER=NO──┘─►◄│
└──────────────────────────────────────────────────────────────────────┘
```

OR

```
┌─ Format ────────────────────────────────────────────────────────────┐
│  ►►──CLASS──CLASSID=AUTOLOG──                                         │
│                              └─SWINDOW=tttt─┘   └─EWINDOW=nnnn─┘──►◄  │
└──────────────────────────────────────────────────────────────────────┘
```

CLASSID

Specify the class ID, or name, by which a class is to be known.

You can choose from two types of class IDs:

Type 1 job classes:

- Must have a single letter or number as the class ID.

- Accept only jobs that have the class number or letter specified as the job class.
- Example: a job class named B is a type 1 class. Only jobs that have B specified as the job class can run as CLASS B jobs.

Type 2 job classes:

- Can have a class ID that is two to eight letters and/or numbers long.
- Accept only jobs that specify the full class ID.
- Example: a job class named NIGHT is a type 2 class. Only jobs that have NIGHT specified as the job class can run as CLASS NIGHT jobs.

SWINDOW and **EWINDOW** (Start and end time of start time window.)
Specify the times that define the start time window of the job class.

Enter the start time in this format: *hhmm*, using the 24-hour time notation system. For example, enter 1700 for 5 p.m., enter 2359 for 11:59 p.m., enter 0300 for 3 a.m.

To specify a 24-hour-a-day start time window, enter 0000 and 2400 as the start time and end time of the window.

The job class start time window serves as a default start time window for jobs assigned to the class. If a job owner submits a job to this class and does not specify a start time window, the job will run under the start time window of the job class. The job class start window is overridden if the job owner does specify a start window.

Specification of the SWINDOW and EWINDOW parameters is optional. The default start time window is open for 24 hours a day.

DEFCPU and **MAXCPU** (Default and maximum limits on use of processor seconds.)
Specify:

1. The default number of processor seconds that jobs of this class may use. The default limit applies to any job that is submitted without specifying a processor seconds limit.

2. The maximum number of processor seconds that jobs of this class may use. The maximum applies to jobs for which the job owner specified a higher limit. The user-specified limit will be reset to the class maximum. For example, if a user specifies a processor seconds limit of 1100 and the class maximum is 1000, the processor seconds limit will be reset to 1000. The user receives a message saying that the change has been made.

The processor seconds limits must be decimal numbers between 1 and 99999999.

You can specify NONE for the maximum value, to allow the possibility of executing jobs without a processor seconds limitation.

Note that only the maximum, and not the default, may be specified as NONE. To execute a job without a processor seconds limit, NONE must be specified to override the defaults when using SUBMIT or CHJOB.

If a job exceeds the limit that applies to it, the DGRIDO exit is called. The default action by this exit is to cancel the job.

DEFPRT and **MAXPRT** (Default and Maximum limits on print records generated.)
Specify:

1. The default number of print records (in thousands) that jobs of this class may generate. The default limit applies to any job that is submitted without specifying a print records limit.

2. The maximum number of print records that jobs of this class may produce. The maximum applies to jobs for which the job owner specified a higher limit. The user-specified limit will be reset to the class maximum. For example, if a user specifies a print record limit of 1100 and the class maximum is 1000, the print record limit will be reset to 1000. The user receives a message saying that the change has been made.

The print record limits must be decimal numbers between 1 and 9999. The limits are in thousands of records. For example, 10 stands for 10,000 records.

You can specify NONE for the maximum value, to allow the possibility of executing jobs without a print record limitation.

Note that only the maximum, and not the default, may be specified as NONE. To execute a job without a print record limitation, NONE must be specified to override the defaults when using SUBMIT or CHJOB.

If a job exceeds the limit that applies to it, the DGRIDO exit is called. The default action by this exit is to cancel the job.

DEFPUN and **MAXPUN** (Default and maximum limits on punch records generated.)
Specify:

1. The default number of punch records (in thousands) that jobs of this class may generate. The default limit applies to any job that is submitted without specifying a punch records limit.

2. The maximum number of punch records that jobs of this class may produce. The maximum applies to jobs for which the job owner specified a higher limit. The user-specified limit will be reset to the class maximum. For example, if a user specifies a punch record limit of 1100 and the class maximum is 1000, the punch record limit will be reset to 1000. The user receives a message saying that the change has been made.

The punch record limits must be decimal numbers between 1 and 9999. The limits are in thousands of records. For example, 10 stands for 10,000 records.

You can specify NONE for the maximum value, to allow the possibility of executing jobs without a punch record limitation.

Note that only the maximum, and not the default, may be specified as NONE. To execute a job without a punch record limitation, NONE must be specified to override the defaults when using SUBMIT or CHJOB.

If a job exceeds the limit that applies to it, the DGRIDO exit is called. The default action by this exit is to cancel the job.

DEFSTOR , **MINSTOR** and **MAXSTOR** (Default, minimum and maximum virtual storage for a job class.)
Specify the range of permissible virtual storage values for this job class. Storage values can be specified in megabytes by specifying 'M', or in kilobytes by specifying 'K' as the terminating character. 'K' is assumed if no terminating character is specified. Consistency in the use of 'K' or 'M' is recommended,

since these values are reported to users in response to a BATCH QUERY CLASS command.

- Enter the amount of virtual storage to be provided to a job of this class by default using the DEFSTOR keyword.

- Enter the minimum amount of virtual storage to be provided to a job of this class using the MINSTOR keyword.

- Enter the maximum amount of virtual storage to be provided to a job of this class using the MAXSTOR keyword.

You can specify an asterisk (*) for DEFSTOR to indicate that by default, jobs of this class run with the CP directory value of virtual storage in the allocated task machine.

You can specify an asterisk (*) for both MINSTOR and MAXSTOR keywords to indicate that no storage requests other than the class default be permitted for jobs of this class. To allow specification of any storage value for a job, both MINSTOR and MAXSTOR must have a value other than an asterisk (*) which is the default.

Note: It is an error to specify an asterisk (*) on only one of MINSTOR or MAXSTOR keywords.

All keywords are optional. If not specified, they take an asterisk (*) which is the default.

The smallest value that may be specified for any keyword is 2048 kilobytes or 2 megabytes. The largest value that may be specified for any keyword is 2096128 kilobytes or 2047 megabytes. A job will be rejected if it requests virtual storage that the task machines are incapable of providing according to their CP directory definitions.

Note: If the VM Batch Facility is installed on VM/ESA 370 Feature, the largest value that can be specified is 16384 kilobytes, or 16 megabytes.

The following table shows the number to enter for various storage sizes:

megabytes	kilobytes
2	2048
4	4096
8	8192
16	16384
32	32768
64	65536
128	131072
256	262144
512	524288
1024	1048576
2047	2096128

STALLDLY and **STALLWRN** (Minimum stall delay and maximum number of stall warnings.)

Specify:

1. The minimum time, in seconds, that jobs in the class may consume no processor time or perform no I/O, before they are deemed to have stalled. At the expiration of this interval, a warning is issued to the job owner. Enter a number from 30 to 9999.

2. The maximum number of stall warnings that may be issued for jobs in the class for any one stall incident, before the DGRIDO exit is called to handle the stall condition. The default action of the exit is to cancel the job. Enter a number from 1 to a maximum of 9999.

   You can specify 0 for STALLWRN to prevent the facility from calling the DGRIDO exit and hence from canceling jobs of this class. Stall warnings are still issued at each stall detection.

   You can specify an asterisk (*) for STALLWRN to prevent the facility from taking any action at all other than the recording of the stall for accounting purposes.

Specification of the STALLDLY and STALLWRN parameters is optional. If not specified, the CONTROL FILE values for MINIMUM-STALL-DELAY and MAXIMUM-STALL-WARNINGS are taken as defaults.

LLSDEF and **LLSOVER** (Subjecting Jobs to LLS.)
Specify:

1. Whether jobs in this class are subject to Load Level Scheduling (LLS) by default. Type LLS to subject the jobs to LLS, type NOLLS to keep jobs independent of LLS activity.

2. Whether individual jobs may override the default LLS subjection for this class. Type YES to allow overrides, type NO to disallow overrides.

Note: The LLSDEF and LLSOVER parameters have no effect if LLS is not active. See "Load Level Scheduling" on page 64 for more information.

Specification of the LLSDEF and LLSOVER parameters is optional. If not specified, LLSDEF is set by default to NOLLS and LLSOVER is set to YES.

AUTOLOG
Define the AUTOLOG job class if you want to allow job owners to have jobs run on their own user IDs. If the AUTOLOG job class is not defined in the CONTROL FILE, the VM Batch Facility will reject jobs submitted with CLASS AUTOLOG specified.

In addition to the class ID, the job class AUTOLOG entry may also specify SWINDOW and EWINDOW values. No other values should be included in the entry. If other values are included, they are ignored.

For more information on entering the start window open and close times, see page 36.

Usage Notes

1. **Job Class A:** The VM Batch Facility uses job class A as the default job class. If a job owner does not specify a job class, the job is automatically assigned to job class A.

   Job class A must always be defined in the VM Batch Facility CONTROL FILE unless you are using only the AUTOLOG job class.

2. **Number of Job Classes You Can Define:** You can define from 1 to a maximum of 256 job classes.

3. **Deleting Job Classes:** Jobs that were already queued when the job class was deleted will be run under the deleted job class.

4. **Changes to job class values:** Jobs already queued when changes were made to a job class will be run under the job class values that were in effect when the jobs were submitted.

5. **Support for VSTOR:** The support for VSTOR will be carried over for this release. If VSTOR is specified, DEFSTOR takes the VSTOR value.

Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits

Changing Maximum Users and Jobs Limits

The BATCH CONTROL SET command cannot change the values of the MAXIMUM-USERS and MAXIMUM-JOBS keywords. To change the these limits, use the following procedure:

1. Use the procedure shown on page 42 to estimate the correct values for MAXIMUM-USERS and MAXIMUM-JOBS.

2. Edit the CONTROL FILE.

   Refer to the *Program Directory* for detailed instructions on how to update the CONTROL FILE.

   Put in the new value(s) for MAXIMUM-USERS and/or MAXIMUM-JOBS. Instructions for setting the values of the keywords appear below.

3. If the new MAXIMUM-JOBS and MAXIMUM-USERS values are higher than the existing values, stop and restart the VM Batch Facility to put the new values into effect.

4. If the new values are lower than the existing values, reduce the number of job owners or queued and running jobs until they meet the new limits. Then stop and restart the VM Batch Facility to put the new limits into effect.

   To reduce the current number of jobs and job owners to meet new limits, we recommend that you wait to make the change until the number of jobs and/or users in progress and/or queued in the VM Batch Facility has fallen to the required levels. If that is not possible, cancel jobs until the limits you require have been reached. Then STOP the VM Batch Facility to prevent new jobs from being received.

   If you restart the VM Batch Facility and the number of jobs or job owners is still above the new keyword values, the VM Batch Facility temporarily sets the keywords to the current number of jobs or job owners and begins operation. The temporary values remain in effect until you stop and restart the VM Batch Facility.

   The VM Batch Facility will send a message to the monitor machine user ID console if this problem occurs.

The VM Batch Facility will ignore changes to MAXIMUM-JOBS and MAXIMUM-USERS if the changes are made with a BATCH CONTROL SET command. The VM Batch Facility will continue to operate using the current MAXIMUM-JOBS and MAXIMUM-USERS values.

Estimating the MAXIMUM-USERS and MAXIMUM-JOBS Limits

The MAXIMUM-USERS and MAXIMUM-JOBS keywords set limits on the number of jobs and job owners that the VM Batch Facility can accept at one time. These keywords determine how many user control blocks and job control blocks the VM Batch Facility will have.

The procedure shown in the box below shows how to estimate the correct limits for your VM Batch Facility, based on the work load you expect. You should also consider the number of task machines available for running jobs when you estimate new values for these limits.

Note: If you increase the limits substantially above the default limits of 50 users and 500 jobs, you may also have to change the CP definition of the VM Batch Facility monitor machine. It may require more virtual storage and more disk storage for the 193 (job file storage) and 194 (control block storage) minidisks. See "Tailoring the Monitor Machine to Handle Higher Work Loads" on page 45 for more information.

Determining Maximum Users and Jobs Limits

1. **Determining the maximum number of users:** Select a number that is at least as large as the average number of VM Batch Facility users at peak times.

   For example, if 80 users are logged on to your system during peak periods, and you expect half of them to use the VM Batch Facility, you should select 40 or a larger number as the maximum number of users.

2. **Determining the maximum number of jobs:** Multiply your estimate of maximum users by an estimate of the number of jobs each user will have in process at one time. If you are not sure of the number to use, we recommend 10 as the average number of jobs each user will have queued and running.

   ```
   Multiply:

       Estimated maximum users
       Estimated queued and running jobs per user
       -----------------------------------------
       Estimated MAXIMUM-JOBS:
                               ------------------
   ```

   In our example, the maximum jobs limit would be 400: 40 users multiplied by 10 queued and running jobs per user.

Filling Out the MAXIMUM-USERS Keyword

Use this keyword to set a limit on the number of job owners that the VM Batch Facility can accept at any one time.

MAXIMUM-USERS specifies the number of user control blocks that the VM Batch Facility will have available for job owners. Every job owner who has a job queued or running must have a user control block.

Changing MAXIMUM-USERS: You cannot use the BATCH CONTROL SET command to put a change in MAXIMUM-USERS into effect. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for instructions on putting a new maximum users limit into effect.

```
 ┌─ Format ─────────────────────────────────────────────┐
 │                                                       │
 │           ┌─MAXIMUM─USERS 50──────┐                   │
 │   ►►──────┤                       ├──►◄               │
 │           └─MAXIMUM─USERS nnnnn───┘                   │
 │                                                       │
 └───────────────────────────────────────────────────────┘
```

nnnnn
  Enter a number from 1 to 50000. The number is the maximum number of job owners that the VM Batch Facility can accept at one time.

Default value	50 job owners.
Range of acceptable values	1 to 50000 job owners.
Recommended value	No recommended value. The correct value depends on the batch processing work load in your system. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for help in calculating MAXIMUM-USERS.

Filling Out the MAXIMUM-JOBS Keyword

Use this keyword to set a limit on the number of jobs that the VM Batch Facility can have queued and running at any one time.

The keyword specifies the number of job control blocks that the VM Batch Facility will have available for jobs. Every job in process in the VM Batch Facility must have its own job control block.

Changing MAXIMUM-JOBS: You cannot use the BATCH CONTROL SET command to put a change in MAXIMUM-JOBS into effect. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for instructions on putting a new maximum jobs limit into effect.

```
┌─ Format ─────────────────────────────────────────────────────────

      ┌─MAXIMUM─JOBS 500───────┐
  ►►──┤                        ├──►◄
      └─MAXIMUM─JOBS nnnnnn────┘

```

nnnnnn
> Enter a number from 1 to 450000. The number is the maximum number of queued and running jobs permitted in the VM Batch Facility at one time.

Default value	500 jobs.
Range of acceptable values	From 1 to 450000.
Recommended value	No recommended value. The correct value depends on the batch processing work load in your system. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for help in calculating MAXIMUM-JOBS.

Tailoring the Monitor Machine to Handle Higher Work Loads

This topic shows the procedures for determining the storage required by the virtual machine that runs the VM Batch Facility monitor, and its 193 and 194 minidisks.

Calculating the Virtual Storage for the Monitor Machine

Use the following formula to calculate the minimum virtual storage required by the virtual machine that runs the VM Batch Facility monitor.

The monitor machine must have at least 2MB (2048K) of virtual storage. The amount of space required for job and user control blocks determines whether additional virtual storage is needed.

```
┌─ Calculating VM Batch Facility Monitor Virtual Storage ─────────────────┐
│                                                                          │
│  Use the formula below to find out how much virtual storage the monitor  │
│  machine will require:                                                   │
│                                                                          │
│          Add:                                                            │
│                                                                          │
│              1,048,576 bytes (1MB)                                       │
│    (64 bytes   X  maximum users)                                         │
│  + (592 bytes  X  maximum jobs)                                          │
│  ------------------------------                                          │
│    Total virtual storage:                                                │
│                        ------                                            │
│                                                                          │
│  If the result is greater than 2MB of storage, use the result as the     │
│  virtual storage size of the monitor machine in the machine's CP         │
│  directory entry.                                                        │
│                                                                          │
│  If the result is less than 2MB, leave virtual storage at 2MB, as        │
│  recommended in the VM Batch Facility installation instructions.         │
│                                                                          │
│  The initial 1024KB of storage is an estimate of the storage required    │
│  for CMS, CMS control blocks, and work space used by the VM Batch        │
│  Facility.                                                               │
│                                                                          │
│  The second and third lines calculate the amount of virtual storage      │
│  needed for user and job control blocks.                                 │
│                                                                          │
│  Example:  If the maximum users limit is 500 and the maximum jobs        │
│  limit is 5000, the monitor virtual machine would require about 4MB of   │
│  storage:                                                                │
│                                                                          │
│          Add:                                                            │
│                                                                          │
│                1,048,576 bytes (1MB)                                     │
│        (64 bytes   X    500)      (maximum users)                        │
│      + (592 bytes  X   5000)      (maximum jobs)                         │
│  ------------------------------                                          │
│    Total virtual storage: 3.85MB of storage                             │
│                        ------                                            │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

Determining the Size of the Monitor Machine 193 Minidisk

Use the instructions in this section to estimate the storage requirements of the monitor machine 193 minidisk. The monitor machine uses the 193 minidisk to store active jobs.

Non recurring jobs are stored by the monitor until they have successfully been started. Recurring jobs are stored until they are flushed.

You may configure a monitor to store jobs in the following ways:

1. If you specify the MAXIMUM-JOB-SIZE keyword in the CONTROL FILE to have a value of zero, then only recurring jobs, and all recurring jobs, are stored on the monitor 193 disk. Recurring jobs must be stored on the monitor 193 disk. If there is no room for them, they are rejected.

   All non recurring jobs are stored in the monitor virtual punch. There are no penalties on jobs that are queued in the punch rather than on the monitor 193 disk. Jobs queued in the punch are eligible to run and subject to VM Batch Facility commands in the same way as jobs queued on the 193 disk.

   In this way, the monitor 193 disk may be reserved for recurring jobs only.

2. If you specify the MAXIMUM-JOB-SIZE keyword in the CONTROL FILE to have a value greater than zero, then all incoming jobs are stored on the monitor 193 disk, except for jobs that exceed the MAXIMUM-JOB-SIZE. These are stored in the monitor virtual punch.

   In this way, the monitor 193 disk may be reserved for all jobs that do not exceed a certain size.

   If a shortage of space arises on the 193 disk, then jobs that would normally be stored on the disk are stored in the punch, except for recurring jobs, which are rejected.

Determining 193 Minidisk Size for all jobs

Use the formula below to estimate the maximum amount of storage needed for the monitor's 193 minidisk when storing all jobs below a certain size.

To estimate the required storage, we recommend that you use the average user and jobs per user estimates of "Filling Out the MAXIMUM-USERS Keyword" on page 43. If you are not sure of the average size of job files, we recommend that you use 50,000 bytes as an estimated job size. That is about 500 lines per job file.

```
        Multiply:

    Estimated average users
    Estimated average number of jobs per user
    Estimated average job size
    ----------------------------------------
    Minimum 193 minidisk storage:
                                -----------
```

If the result indicates that more than 10 cylinders of IBM 3380 DASD storage (or equivalent) are required, redefine the 193 minidisk using the new storage figure. Ten cylinders is the size recommended for the 193 minidisk in the VM Batch Facility installation instructions.

> **Determining 193 Minidisk Size for Recurring Jobs Only**
>
> Use the formula above, but use the estimated number of users that you expect
> to submit recurring jobs, and the estimated number of recurring jobs per user.

Determining the Size of the Monitor Machine 194 Minidisk

The VM Batch Facility uses the monitor machine 194 minidisk to store the BATCH JOBS, BATCH USERS, BATCH MACHINES, and BATCH SYSTEM files. These files record the contents of the virtual storage control blocks used by the VM Batch Facility when it is operating.

You should not have to increase the storage required for the 194 minidisk unless you increase MAXIMUM-JOBS to more than 1500, and MAXIMUM-USERS to more than 1000.

Use the procedure in the following box to determine whether the 194 minidisk needs more storage.

```
┌─── Determining 194 Minidisk Size ───────────────────────────────┐
│                                                                 │
│ Use the following formula to estimate the storage needed for the monitor
│ machine 194 minidisk:                                           │
│                                                                 │
│         Add:                                                    │
│                                                                 │
│    MAXIMUM-JOBS keyword value  X 536 bytes                      │
│    MAXIMUM-USERS keyword value X  64 bytes                      │
│    ----------------------------------------                     │
│    Minimum 194 minidisk storage:                               │
│                                  ---------                      │
│                                                                 │
│ If the result indicates that more than 2 cylinders of IBM 3380 DASD storage (or
│ equivalent) are required, increase the size of the 194 minidisk.
│                                                                 │
│ Two cylinders is the amount of storage recommended for the monitor machine
│ in the sample directory entry shown in "How the Monitor Virtual Machine is
│ Defined" on page 16.                                            │
│                                                                 │
└─────────────────────────────────────────────────────────────────┘
```

Identifying Authorized Users

Use the ADMIN keyword to identify authorized users to the VM Batch Facility. Authorized users can enter the privileged forms of VM Batch Facility commands to operate and administer the VM Batch Facility. (See Chapter 4, "Administering and Operating the VM Batch Facility" on page 71 for descriptions of these commands.)

```
┌─── Format ──────────────────────────────────────────────────────┐
│                                                                  │
│  ►►──ADMIN──userid───────────────►◄                              │
│                      └─nodeid─┘                                  │
│                                                                  │
└──────────────────────────────────────────────────────────────────┘
```

userid

   Enter the user ID of the person you would like to designate as an authorized user of the VM Batch Facility.

   To designate more than one authorized user, add a separate ADMIN keyword to the CONTROL FILE for each authorized user. Each keyword must be on a separate line.

nodeid

   Enter the node ID where the user ID is located (optional).

   The node ID is only required when the user to be authorized is on a remote node. Note that the first ADMIN statement must declare an authorized user on the same node as the monitor.

Default value	You must add at least one ADMIN keyword to the CONTROL FILE before you can run the VM Batch Facility.
Maximum number of authorized users	16 authorized users.

Usage Notes

1. **Limiting command authorization:** You can limit the access that authorized users have to privileged commands by altering the VM Batch Facility command screening exit (DGRUCD EXEC). See "Command Screening Exit (DGRUCD EXEC)" on page 138 for more information on the command screening exit.

   If you do not alter the exit, any authorized user who is logged on can enter any privileged command.

2. **Adding authorized users:** See "Enabling Users to Use the VM Batch Facility" on page 74 for additional information on adding authorized users.

Specifying the Default Alert Message Receiver

To designate a default alert message receiver, ensure that the user ID of the receiver is identified in the first ADMIN entry in the CONTROL FILE.

When the VM Batch Facility is started, it designates the first ADMIN user ID in the CONTROL FILE as the alert message receiver. The alert message receiver must be a user ID on the same node as the monitor. The receiver can be changed with the BATCH ALERT command, but whenever the VM Batch Facility is stopped and restarted or the BATCH CONTROL SET command is entered, the receiver will be reset to the first ADMIN user.

In the sample CONTROL FILE below, user OPERATOR is the default alert message receiver:

```
┌─── Sample CONTROL FILE:  Default Alert Message Receiver ───────
*  CONTROL FILE
     .
     .
     .
* Default alert message receiver
ADMIN OPERATOR LOCALNODE
* Other authorized users
ADMIN OTHERUSR OTHERNODE
ADMIN AUTHUSER LOCALNODE
```

Adjusting the Length of Time That Jobs Can Remain Stalled

The MINIMUM-STALL-DELAY and MAXIMUM-STALL-WARNINGS keywords determine together, how long a job can remain stalled before the monitor takes action to resolve the stall. These keywords provide default values that may be overridden by individual CLASS statements.

A job is stalled if it has not used the processor nor performed any I/O during the time specified in the MINIMUM-STALL-DELAY entry. If a stalled job is detected, the VM Batch Facility sends out stall warning messages to the job owner. It waits for the interval specified in MINIMUM-STALL-DELAY before sending out any subsequent warning messages.

If the stalled job has not restarted when the last stall warning message is sent, the VM Batch Facility calls the DGRIDO exit to handle the condition. The default action by this exit is to cancel the job.

To estimate the length of time that a job can remain stalled before action is taken, use the formula below:

```
┌─── Calculating How Long Jobs Can Remain Stalled ──────────────────┐
│                                                                   │
│        Multiply:                                                  │
│                                                                   │
│   MINIMUM-STALL-DELAY value                                       │
│   (MAXIMUM-STALL-WARNINGS +1)                                     │
│   ----------------------------                                    │
│   Maximum stall time:                                             │
│                       --------                                    │
│                                                                   │
│  For example, if MAXIMUM-STALL-WARNINGS is set to 5 and           │
│  MINIMUM-STALL-DELAY is 60 seconds, a job could remain stalled    │
│  for a total time of 6 minutes before the exit is called.         │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

If a stalled job restarts before the maximum time has expired, the count of stall warnings is reset to zero.

Note: The MINIMUM-STALL-DELAY and MAXIMUM-STALL-WARNINGS keywords are used to define the stall activity facility wide. The global stall activity can be redefined at Class level, by specifying the parameters STALLDLY and STALLWRN for the CLASS keyword in the CONTROL FILE. See "Filling Out the CLASS Keyword" on page 35 for additional information on defining stall activity at class level.
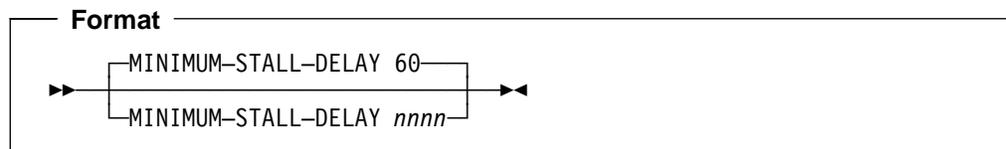
Unlike the specification at the CLASS level, the global MAXIMUM-STALL-WARNINGS may not be specified as 0 or an asterisk (*).

Specifying the Time Interval for Detecting Stalled Jobs

Use the MINIMUM-STALL-DELAY keyword to specify the number of seconds that a job can be stalled before the VM Batch Facility sends out a stall warning message about the job.

The VM Batch Facility determines when a job is stalled by checking the resource usage of the task machine running the job. A stalled job is one that has not used the processor nor performed any I/O during the time that you specify with this keyword.

Note: The MINIMUM-STALL-DELAY keyword defines the stall activity facility wide. The global value can be redefined at Class level, by specifying the parameter STALLDLY for the CLASS keyword in the CONTROL FILE. See "Filling Out the CLASS Keyword" on page 35 for additional information on defining stall activity at class level.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│            ┌─MINIMUM─STALL─DELAY 60───┐                            │
│  ►►────────┤                          ├────►◄                      │
│            └─MINIMUM─STALL─DELAY nnnn──┘                           │
└───────────────────────────────────────────────────────────────────┘
```

nnnn
Enter a number from 30 to 9999. The number is the number of seconds of real time (clock time) that must elapse without processor usage or I/O activity for a job to be considered stalled.

Note: If you set a limit greater than 15 minutes, the VM Control Program may log off a stalled task machine before the VM Batch Facility sends a stall warning message. VM cancels a stalled job only if the job has stalled because of a pending read.

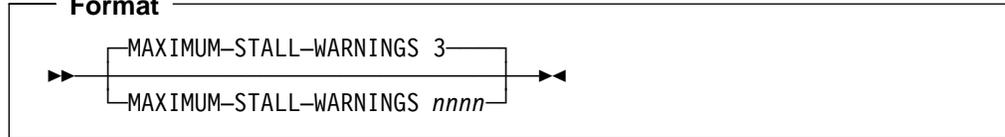|	
Default value	60 seconds. A job is considered stalled if it has not used the processor or performed I/O during a time of at least 1 minute.
Recommended value	60 seconds.
	Setting a lower value may cause the VM Batch Facility to send out stall warnings for jobs that are not permanently stalled. Jobs may appear to stall when:
	• System work loads are high and response times are long. • A tape mount or similar task is being performed.
	A higher value may let jobs remain stalled for too long without warning messages being sent.
Range of acceptable values	30 to 9999 seconds.

Specifying the Number of Stall Warnings to Be Issued

Use the MAXIMUM-STALL-WARNINGS keyword to specify the number of times the VM Batch Facility will send out a message warning that a job has stalled. If more than one stall warning is sent, the VM Batch Facility waits for the interval specified in the MINIMUM-STALL-DELAY keyword between messages.

A stalled job is one that has not used the processor nor performed I/O longer than the MINIMUM-STALL-DELAY period. If a stalled job does not restart before the time limits and warnings have run out, then the monitor calls the DGRIDO exit to handle the stall. The default action of this exit is to cancel the job.

Note: The MAXIMUM-STALL-WARNINGS keyword defines the stall activity facility wide. The global value can be redefined at Class level, by specifying the parameter STALLWRN for the CLASS keyword in the CONTROL FILE. See "Filling Out the CLASS Keyword" on page 35 for additional information on defining stall activity at class level.

```
┌─── Format ──────────────────────────────────────────────────┐
│                                                              │
│              ┌─MAXIMUM─STALL─WARNINGS 3────┐                 │
│      ►►──────┤                             ├──►◄             │
│              └─MAXIMUM─STALL─WARNINGS nnnn──┘                │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

nnnn
   Enter a number from 1 to a maximum of 9999. The number is the number of times that the VM Batch Facility will send a message warning of a stalled job before taking action to resolve the stall.
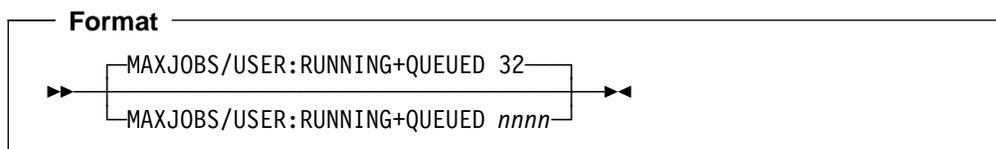
Default value	3 stall warning messages.
Recommended value	3 stall warning messages.
	Fewer warnings may cause jobs to be canceled before the job owner has a chance to respond.
	More warnings may delay the cancellation of a stalled job longer than is necessary.
Range of acceptable values	1 to 9999 warnings.

Limiting the Number of Jobs Each User Can Submit

The following keywords are used to limit the number of jobs that each user can submit at any one time.

Limiting Running and Queued Jobs Belonging to One User

Use the MAXJOBS/USER:RUNNING+QUEUED keyword to specify the maximum number of jobs that one job owner can have queued and running at any one time. The VM Batch Facility will reject additional jobs from a job owner who has reached the limit. The job owner is allowed to submit additional jobs only after at least one of his or her jobs is completed or canceled.

```
┌─ Format ──────────────────────────────────────────────────────────┐
│                                                                    │
│          ┌─MAXJOBS/USER:RUNNING+QUEUED 32──┐                        │
│   ►►─────┤                                 ├────►◄                  │
│          └─MAXJOBS/USER:RUNNING+QUEUED nnnn─┘                       │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```

nnnn

Enter a number from 1 to a maximum of 9999. The number will be the maximum number of queued jobs and running jobs that any user can have in the VM Batch Facility at one time.

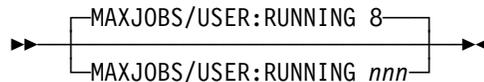|	
Default value	32 queued and running jobs.
Recommended value	No recommended value.
	The number you select should reflect the number of batch users you have, the average number of jobs each user has in process in the VM Batch Facility at one time, and the resources (task machines, user and job blocks, minidisk space) you have given to the VM Batch Facility.
Range of acceptable values	1 to 9999 jobs.

Usage Note

CLASS AUTOLOG jobs: CLASS AUTOLOG jobs count toward the MAXJOBS/USER:RUNNING+QUEUED limit only while they are queued. They are not counted once they begin to run.

Limiting Running Jobs Belonging to One User

Use the MAXJOBS/USER:RUNNING keyword to specify the maximum number of jobs that one job owner can have running in the VM Batch Facility at any one time. A job is running when a task machine is executing it.

No additional jobs belonging to a user who has reached this limit will be started until at least one of the user's jobs has been completed or canceled.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│                                                                   │
│            ┌─MAXJOBS/USER:RUNNING 8─┐                             │
│   ►►───────┤                        ├───►◄                        │
│            └─MAXJOBS/USER:RUNNING nnn─┘                            │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

nnn
>   Enter a number from 1 to 128. This is the maximum number of jobs that a job owner can have running in the VM Batch Facility at one time.

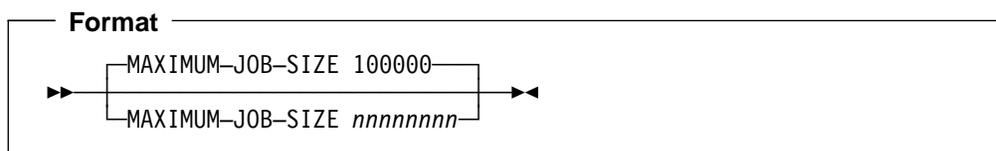|	
Default value	8 jobs running at one time.
Recommended value	No recommended value.
	The value you choose should reflect the number of task machines available to the VM Batch Facility and to each job class. If MAXJOBS/USER:RUNNING is higher than the number of task machines available to a class, for example, one or a few users may monopolize the available task machines.
Range of acceptable values	1 to a maximum of 128.

Usage Notes

1. **Limited by the number of task machines available:** The number of jobs a user can have running may also be limited by the number of VM Batch Facility task machines. If the VM Batch Facility has three task machines, no user can have more than three jobs running at once, regardless of the MAXJOBS/USER:RUNNING value.

2. **CLASS AUTOLOG jobs:** A job owner's CLASS AUTOLOG jobs do not count toward the MAXJOBS/USER:RUNNING limit.

Specifying the Maximum Size of Job Files on the 193 Minidisk

Use the MAXIMUM-JOB-SIZE keyword to limit the size of the non-recurring job files that the monitor machine will store on its 193 minidisk; or to reserve the monitor's 193 disk for recurring jobs exclusively. Recurring jobs must be stored on the 193 disk; if there is no room for them they are rejected. Non-recurring jobs may be stored on the 193 disk or in the monitor machine virtual punch. If the 193 disk becomes full, all non-recurring jobs are queued in the punch. Setting an upper limit on job files prevents extremely large files from filling the 193 minidisk.

There are no penalties on jobs that are queued in the punch rather than on the 193 minidisk. Jobs in the punch queue are eligible to run and are subject to the VM Batch Facility commands.

```
┌─ Format ────────────────────────────────────────────────────────
│              ┌─MAXIMUM–JOB–SIZE 100000─┐
│   ►►─────────┤                         ├──────►◄
│              └─MAXIMUM–JOB–SIZE nnnnnnnn─┘
└──────────────────────────────────────────────────────────────────
```

nnnnnnnn

Enter the maximum file size in bytes. The number you enter must be a decimal number from 0 to 99999999.

The VM Batch Facility includes all appended files in determining the size of a job file.

Non-recurring job files that are larger than the limit you enter will be queued in the monitor machine punch instead of on the 193 minidisk. There is no limit on job size applied to recurring jobs other than the capacity of the monitor's 193 minidisk.

Setting the size to **0** causes the VM Batch Facility monitor to queue all non-recurring jobs in the monitor machine punch. This effectively restricts the jobs stored on disk to recurring jobs only.

Recurring jobs changed to non-recurring jobs by CHJOB command remain on the disk until they run.

Default value	100000 bytes.
Recommended value	100000 bytes.
	You may wish to adjust the value to reflect the size of the 193 minidisk and the number of large jobs submitted to the VM Batch Facility.
Range of acceptable values	0 to 99999999 bytes.

Specifying the Type of Accounting Support

Use the ACCOUNT keyword to indicate the type of job accounting assistance the VM Batch Facility should provide.

```
┌─ Format ────────────────────────────────────────────────┐
│                                                          │
│              ┌─ACCOUNT─0─┐                               │
│         ►►───┤           ├───►◄                          │
│              └─ACCOUNT───┬─1─┬─┘                         │
│                          ├─2─┤                           │
│                          └─3─┘                           │
│                                                          │
└──────────────────────────────────────────────────────────┘
```

0

Enter "0" to have the VM Batch Facility provide no accounting assistance. The only job accounting data you will have will be standard VM records of connect time and resource usage of task virtual machines. All costs will be assigned to the task machines.

The ACCOUNT keyword defaults to 0 if no value is specified for it.

1

Enter 1 to call the VM Batch Facility job accounting exit (DGRUAC EXEC) whenever a job is completed or canceled. The exit is provided with information relevant to the amount of system resource that was used by each job.

It can be modified to generate accounting records in any form using part or all of the job information supplied.

As shipped with the product, it generates an accounting record for each job that is completed.

The accounting exit stores accounting records on the monitor machine 195 minidisk. The file containing the records is called DGRUAC RECORDS.

For more information on the DGRUAC exit and for a sample accounting record, see "Job Accounting Exit (DGRUAC EXEC)" on page 135.

2

Enter 2 to record job owner account information in the standard VM record of task machine connect time and resource usage. The VM record created when a job is completed will charge the task machine connect time and resources to the account number or name specified by the job owner.

3

The VM Batch Facility performs both of the accounting actions described in options 1 and 2.

Default value   Option 0–Only VM accounting is performed; the accounting exit (DGRUAC EXEC) is not called to store accounting information.
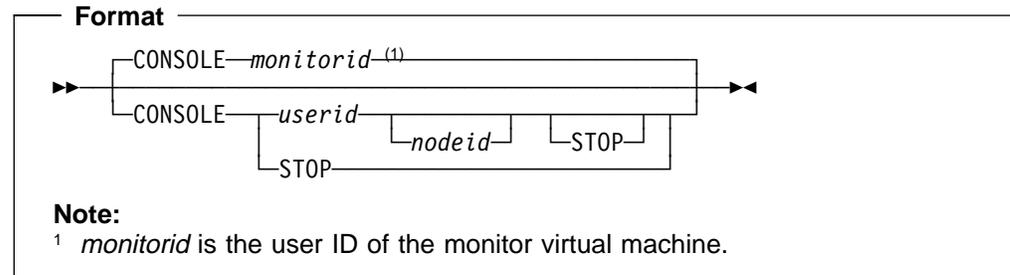
Usage Notes

 1. **CLASS AUTOLOG job accounting**: The VM Batch Facility does not keep accounting records on CLASS AUTOLOG jobs. The cost of running a CLASS AUTOLOG job is charged to the job owner's user ID using VM accounting facilities.

2. **Accounting minidisk automatically cleared:** The VM Batch Facility accounting exit (DGRUAC EXEC) stores the accounting records it generates on the monitor machine 195 minidisk. To avoid having the VM Batch Facility shut down when the minidisk is full, the accounting exit does the following:

   a. Sends alert messages when the minidisk becomes 85% and 90% full.

   b. Sends the DGRUAC RECORDS file to the alert message receiver if the minidisk becomes 95% full. The exit also erases the file from the 195 minidisk.

   The 195 minidisk is used only to hold accounting records; no other VM Batch Facility files are stored there.

Specifying the Disposition and Routing of Console Messages

Use the CONSOLE keyword to specify the default state of console message spooling. The default state of console message spooling will take effect whenever the VM Batch Facility is started or the BATCH CONTROL SET command is entered. The CONSOLE keyword resets any BATCH CONSOLE commands that have been entered.

```
┌─ Format ──────────────────────────────────────────────────────┐
│                                                                │
│          ┌─CONSOLE──monitorid──(1)─────────────────────┐       │
│  ►►──────┤                                             ├──►◄   │
│          └─CONSOLE──┬─userid──┬─────────┬─┬──────┬─────┘       │
│                     │         └─nodeid──┘ └─STOP─┘             │
│                     └─STOP────┘                               │
│                                                                │
│  Note:                                                         │
│  ¹  monitorid is the user ID of the monitor virtual machine.   │
└────────────────────────────────────────────────────────────────┘
```

userid

To designate a default console message receiver, enter a user ID after the CONSOLE keyword. When the VM Batch Facility is started or a BATCH CONTROL SET command is entered, console messages will be spooled to the user ID you specify.

nodeid

The node of the default console message receiver. This need only be entered if the console receiver is on a remote node.

STOP

Enter STOP after the CONSOLE keyword if you want the VM Batch Facility to record console messages only on command. You will have to enter the BATCH CONSOLE START command or change the CONSOLE keyword to have the VM Batch Facility start recording console messages.

user ID node ID STOP

To designate a default message receiver but have console message spooling inactive when the VM Batch Facility starts, enter both a user identification and STOP.

When a BATCH CONSOLE START command is entered, the VM Batch Facility will start spooling console messages to the user ID identified in the CONSOLE entry.

Default value   The console is spooled to the VM Batch Facility monitor machine itself.

Usage Notes

1. **user ID checked:** Be sure that any user ID that you specify is valid for the system. If it is not a valid user ID, the console is spooled to the VM Batch Facility monitor machine itself.

2. **Clearing the console log file:** Because console message spooling uses system spool space, be sure to close the file and process it before it uses too much spool space. Use the BATCH CONSOLE CLOSE command to close the file.
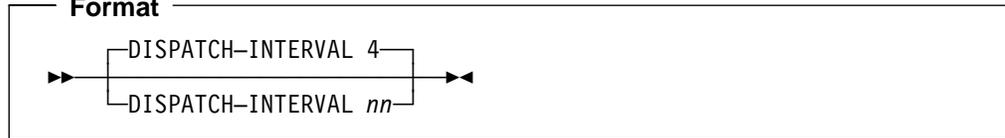
3. **Relation to the BATCH CONSOLE command:** The CONSOLE keyword sets the default status of console message spooling. While the VM Batch Facility is operating, you can change the status of spooling with the BATCH CONSOLE command. When the VM Batch Facility is stopped and restarted, however, console message spooling will be reset to the status recorded in the CONSOLE keyword. For more information on the BATCH CONSOLE command, see "BATCH CONSOLE: Controlling Console Message Recording" on page 91.

Adjusting the Dispatch Interval

Use the DISPATCH-INTERVAL keyword to specify the maximum length of time that the VM Batch Facility should wait before starting jobs.

The VM Batch Facility looks for task machines that are available for running jobs whenever the DISPATCH-INTERVAL elapses, or whenever a job is submitted or ended.

```
┌─ Format ──────────────────────────────────────────────┐
│            ┌─DISPATCH─INTERVAL 4─┐                     │
│   ►►───────┤                     ├──────►◄             │
│            └─DISPATCH─INTERVAL nn─┘                    │
└───────────────────────────────────────────────────────┘
```

nn

   Enter a number between 4 and 30. The number is the maximum number of seconds that should elapse before the VM Batch Facility starts new jobs in available task machines. The interval is measured in real clock seconds.

Default value	4 seconds
Recommended value	4 seconds
	Setting a longer dispatch interval may decrease the amount of system resources that the monitor machine uses. However, it can also result in delays in starting jobs.
Range of acceptable values	4 to 30 seconds.

Adjusting the Check-Running-Jobs Interval

Use the CHECK-RUNNING-JOBS keyword to specify the maximum amount of time that the VM Batch Facility should wait before checking the resource usage of jobs that are running.

The VM Batch Facility checks resource usage to detect stalled jobs and jobs that have exceeded their processor seconds, print, or punch limits.

```
 ┌─ Format ─────────────────────────────────────────────────
 │         ┌─CHECK─RUNNING─JOBS 10─┐
 ►►────────┤                       ├────────►◄
           └─CHECK─RUNNING─JOBS nn─┘
```
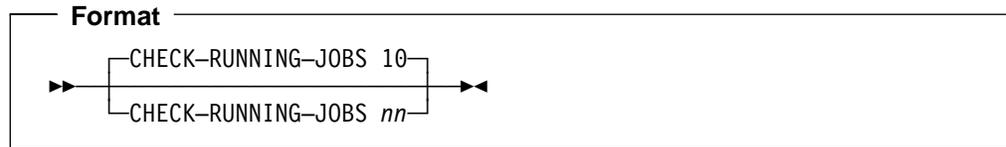
nn
   Enter a number from 4 to 99. The number is the maximum number of seconds that the VM Batch Facility will wait before checking the resource usage of running jobs. The number represents seconds of clock time (real time).

   Be sure that the interval you specify is shorter than the MINIMUM-STALL-DELAY interval. Otherwise, stall warning messages will be repeated at longer intervals than the MINIMUM-STALL-DELAY interval, and cancellation of stalled jobs will be delayed.

Default value	10 seconds.
Recommended value	10 seconds.
	A shorter interval may let the VM Batch Facility detect jobs that are stalled or over resource limits more quickly. But the monitor machine may also use more processor time in your system.
	A longer interval may reduce VM Batch Facility consumption of processor time in your system. However, it may also delay the cancellation of jobs that have stalled or that are over resource limits.
Range of acceptable values	4 to 99 seconds.

Adjusting the Job Initialization Time

Use the JOB-INIT-TIME/100000-RECS keyword to specify the maximum amount of time, per 100000 records, that the VM Batch Facility should allow for a task machine to read in a job.

The VM Batch Facility cancels a job that exceeds this time to initialize.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│                                                                   │
│            ┌─JOB─INIT─TIME/100000─RECS 4─┐                         │
│  ▶▶────────┤                             ├────▶◀                   │
│            └─JOB─INIT─TIME/100000─RECS nn─┘                        │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

nn
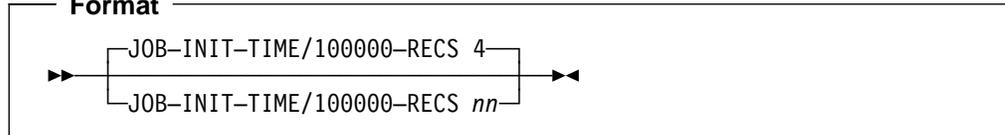
Enter a number from 4 to 99. The number is the maximum number of minutes that the VM Batch Facility will wait, per 100000 records, for a job to initialize.

Note that the 100000 records referred to are spool records generated by the CMS DISK DUMP command invoked to pass appended files to the VM Batch Facility, and not the number of records in the CMS file. For example, 500 records in a CMS file with an 80-byte record length generates 800 spool records.

Default value	4 minutes.
Recommended value	4 minutes.
	A longer interval will allow more time for very large jobs to be loaded.
Range of acceptable values	4 to 99 minutes.

Load Level Scheduling

You can automatically control the resources used by the VM Batch Facility by activating the Load Level Scheduler (LLS) of the VM Batch Facility.

When active, the LLS keeps a running average of the number of virtual machines in the CP queues that are waiting for each of the following system resources:

- Processor
- Paging
- I/O
- Storage.

This running average is called the Measured Contention Value (MCV) of the resource.

You activate the LLS by defining threshold values for the MCV for each of the preceding resources:

- **The Suspend Contention Value (SCV)** — when the Measured Contention Value for a resource exceeds the Suspend Contention Value, contention for the resource is said to exist, and the LLS acts to reduce the consumption of that system resource by the VM Batch Facility.

- **The Resume Contention Value (RCV)** — after reducing the VM Batch workload because of contention for a resource, when the Measured Contention Value for a resource falls below the Resume Contention Value, the LLS acts to restore the VM Batch Facility workload.

Jobs can be defined as being subject to LLS activity, or independent of it. The LLS has no effect on jobs defined as independent of its control.

Defining Load Level Scheduler Activity

Initial LLS activity is defined by coding LLS statements in the Control File. One statement is required for each of the system resources (processor, paging, I/O, storage) that is to be monitored by the LLS. This definition sets the LLS activity of the Facility each time the monitor is started, or a BATCH CONTROL SET command is entered. If there is no LLS statement in the Control File for a resource, then the LLS is not activated for that resource. Refer to "Filling Out the LLS Keyword" on page 69 for more information.

LLS activity for a resource may be dynamically reviewed, or changed, or stopped, while the monitor is running, by entering the BATCH LLS command. Refer to "BATCH LLS: Scheduling LLS Activity" on page 106 for more information.

Defining Job Subjection to Load Level Scheduler Activity

The subjection of jobs to LLS activity is defined when coding the CLASS statements in the Control File. You define whether jobs in each CLASS are to be subject to LLS activity by default (the LLSDEF parameter), and whether individual jobs may override the default subjection (the LLSOVER parameter). Refer to "Filling Out the CLASS Keyword" on page 35 for more information.

Individual jobs should be submitted to an appropriate Class to have the required LLS subjection. However, if the Class definition permits, the default LLS subjection

of a job may be overridden by specifying the LLS or NOLLS options on the BATCH
SUBMIT and BATCH CHJOB commands.

Both queued and running jobs may have their LLS subjection changed by entering
the BATCH CHJOB command with the LLS or NOLLS option specified as required.
Authorized users may change the LLS subjection of any job unconditionally, and
unauthorized users may only change their own jobs, and then only if the change is
within the scope of the Class override definition. If a job has been suspended by
the LLS, then an authorized user may enter a BATCH RESUME command to
resume a job immediately, and to prevent it from suffering any later LLS
suspension.

Note: Jobs that use VMCF, or steal the External new PSW, or alter the CMS
environment in any way, may not respond to LLS suspension or resumption. Such
jobs should not be subjected to LLS activity, as it is possible that they may be
suspended by the LLS and never resumed.

Class AUTOLOG jobs are not subject to LLS activity.

How Subject Jobs Are Affected by LLS Activity

When you set the LLS active for a resource by way of the LLS Control File
statement or the BATCH LLS command, you specify:

1. **The Suspend Contention Value (SCV)** — The SCV defines the value of the
   MCV that will constitute a state of contention for which LLS action is required.

   When the Measured Contention Value for a resource exceeds the Suspend
   Contention Value, no new LLS subject jobs will be dispatched; and currently
   running LLS subject jobs are suspended, one every 20 seconds, until the MCV
   falls below the SCV, or until there are no more LLS subject jobs running.

2. **The Resume Contention Decrement (RCD)** — The RCD specifies the
   amount that the SCV is decremented by to compute the **Resume Contention
   Value (RCV)**. The RCV defines the value of the MCV at which the LLS will act
   to resolve a state of contention by restoring the workload of the VM BATCH
   Facility.

   When the Measured Contention Value for a resource falls below the Resume
   Contention Value, and contention does not exist for any other active resource,
   jobs suspended by the LLS are resumed, one every 20 seconds, until the MCV
   rises above the RCV, or all LLS suspended jobs have resumed.

   When all LLS suspended jobs have resumed, and the MCV remains below the
   RCV for all active resources, then all queued LLS subject jobs that have not
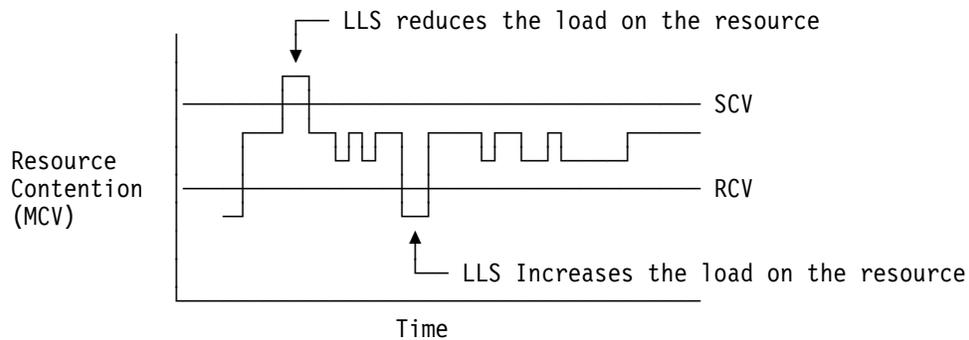   been dispatched because of contention, become available for dispatch.

Figure 2. Example of LLS Adjusting Workload

The current state of the LLS may be reviewed by entering the BATCH LLS command with no parameters.

Job owners are not notified when the LLS takes any action on a job. However, each LLS action is logged to the monitor console.

Jobs manually suspended by using the BATCH SUSPEND command will not be resumed by the LLS when resolving contention.

Tuning LLS Activity

It is not possible to recommend values for the LLS thresholds. If you choose to activate the LLS, you will have a reason for doing so. Choosing threshold values will depend on the load on your system at various times.

Finding suitable LLS thresholds will initially be a matter of trial and error. The following technique is recommended:

- Do not set any LLS values in the Control File. Let the monitor initialize without any LLS activity.

- Get an initial idea of your system's queues by entering the CP IND QUEUES command.

- If there is no job activity within the Facility, then submit a dummy job to the monitor, perhaps with DATE HOLD, to ensure that the monitor is active.

- Activate the LLS for your chosen resources dynamically, using the BATCH LLS command. Select a very high value for the SCV, so that effectively LLS contention will never exist.

- Periodically, or when necessary, enter the BATCH LLS command with no parameters. This will return to your terminal a message describing the status of the LLS activity for each resource; in particular, the MCV as measured by the Facility will be displayed. This will give you a good idea of the MCV's that the Facility is measuring, and will serve as a guide for determining your threshold values.

  Note: The MCV is calculated as a running average over a period of about 15 seconds. When the data from which the MCV is calculated has not been collected for long enough to give a valid MCV, such as shortly after monitor initialization, the MCV takes the value -1 to show that it is not yet valid.

  Also note that the monitor itself is dormant when there are no jobs queued or running. In this dormant state it does not collect the data to calculate the MCV,

so unless there are jobs pending or running under the monitor the MCV will have the value -1.

- Gradually decrease the SCV until you find the LLS acting according to your requirements.
- Set suitable initial values in the Control File when you have found your levels.

The Suspend Contention Value should be set to reflect a queue depth that is unacceptable in your environment. Too low a Suspend Contention Value can cause the unnecessary suspension of jobs, reducing the throughput of the VM Batch Facility. Too high a Suspend Contention Value can prevent LLS from adjusting the VM Batch Facility workload to alleviate resource contention.

The Resume Contention Decrement should be set to a reflect a queue depth that is acceptable in your environment. Too high a Resume Contention Decrement can cause jobs to remain suspended after the system load falls and resources become available. Too low a Resume Contention Decrement can cause jobs to be resumed before the system has sufficient resources available, and in some cases can cause thrashing.

Thrashing occurs when the LLS is continuously suspending and resuming jobs. For example, suppose that the SCV and RCD are set so that when a particular job is suspended, the Measured Contention Value falls below the Resume Contention Value. The LLS therefore resumes the job, but as soon as it is resumed, the Measured Contention Value exceeds the Suspend Contention Value, and the job will once again be suspended; and so on.

The LLS can detect thrashing and attempts to resolve it automatically. It computes a new Resume Contention Value by subtracting the Resume Contention Decrement from the current Resume Contention Value. The Facility also issues a message to the alertee notifying the detection of LLS thrashing.

Although thrashing will eventually be resolved by this mechanism, it can have a severe effect on Facility throughput, and should be prevented by adjusting the LLS definitions in the Control File.

How Jobs Are Selected for Suspension and Resumption

The VM Batch Facility monitors the rate of consumption of the four LLS system resources (processor, paging, I/O, and storage) of every running job.

Suspension: When a job is to be suspended by the LLS because of excessive contention for a resource, it is selected in the following way.

1. If more than one active resource is under contention, then the resource with the highest value of the ratio (MCV-SCV)/SCV is selected.

2. The running LLS subject job that was the highest consumer of this selected resource during the last CHECK-RUNNING-JOBS interval is suspended.

The resource for which the job is suspended is recorded for later use in resume processing.
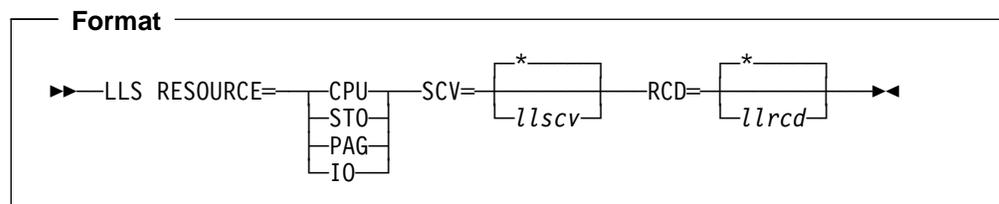
Resumption: When a job is resumed by the LLS after contention, it is selected in the following way.

1. If more than one active resource was under contention, then the resource with the highest value of the ratio (RCV-MCV)/RCV is selected.

2. The LLS suspended job that has the smallest recorded consumption of this selected resource (as measured for the last CHECK-RUNNING-JOBS interval before it was suspended) is resumed, providing that it was suspended for this resource.

3. If no job is resumed for this resource, then the next resource is selected, until a job is resumed.

Filling Out the LLS Keyword

Use the optional LLS keyword to define the Load Level Scheduler threshold values for each of the LLS monitored system resources.

```
┌─ Format ──────────────────────────────────────────────────────────────┐
│                                        *                 *             │
│  ►►──LLS RESOURCE=──┬─CPU─┬──SCV=──┬───────┬──RCD=──┬───────┬──►◄       │
│                     ├─STO─┤        └─llscv─┘        └─llrcd─┘           │
│                     ├─PAG─┤                                            │
│                     └─IO──┘                                            │
└───────────────────────────────────────────────────────────────────────┘
```

RESOURCE

  Type the identifier for the required system resource:

  CPU CPU
  STO Storage
  PAG Paging
  IO   I/O

SCV

  llsscv must be specified as positive integer greater than zero. A specification of an asterisk (*) deactivates all LLS activity for the selected resource.

RCD

  llsrcd must be specified as positive integer greater than zero. *llsrcd* must be less than *llsscv* . A specification of an asterisk (*) deactivates all LLS activity for the selected resource.

A separate specification of the LLS keyword is required for each resource. If no LLS keyword is found in the Control File for a resource, LLS is deactivated for the resource.

Default value	**SCV**	*
	RCD	*
Recommended value	No recommendation.	
	The values for SCV and RCD depend on your system load and behavior.	
	• Too large a value of SCV may prevent the VM Batch workload from decreasing sufficiently to alleviate system contention.	
	• Too small a value of SCV may cause the VM Batch workload to decrease too much, with a corresponding effect on VM Batch throughput.	
	• Too small a value of RCD may cause jobs to be resumed before the system is ready, resulting in thrashing of LLS suspend/resume activity.	
	• Too large a value of RCD may prevent jobs from resuming after LLS suspension.	
Range of acceptable values	**SCV**	From 1 to 99999999.
	RCD	From 1 to 99999999.
	Note: A specification of an asterisk (*) for either SCV or RCD values deactivates all LLS activity for the selected resource.	

LLS Keyword

Chapter 4. Administering and Operating the VM Batch Facility

This chapter explains how to use the VM Batch Facility authorized user commands to administer and operate the VM Batch Facility. The tasks that you can perform are listed below. For help with the tasks, see the page numbers shown in the list.

- Starting the VM Batch Facility. See page 72.

- Stopping the VM Batch Facility. See page 117.
  - Interrupting jobs that are executing
  - Waiting until jobs that are executing are done.

- Enabling users to use the VM Batch Facility. See page 74.
  - General users
  - Authorized users
  - Users of Remote Job Entry (RJE) workstations.

- Changing Monitor and Task Machine Passwords. See pages 77 and 78.

- Using VM Batch Facility Accounting Facilities. See page 80.

- Monitoring the VM Batch Facility:
  - Finding out which user is the current alert message receiver and directing alert messages to a new receiver (BATCH ALERT command). See page 81.

  - Starting or stopping recording of monitor machine console messages (BATCH CONSOLE command). See page 91.

  - Finding out which user is the current console message receiver and directing console messages to a new receiver (BATCH CONSOLE command). See page 91.

  - Sending the console message log file to the console message receiver (BATCH CONSOLE command). See page 91.

- Controlling jobs:
  - Getting information on jobs (BATCH STATUS command). See page 116.

  - Increasing the priority of a job (BATCH PRIORITY command). See page 108.

  - Suspending and resuming the execution of a job (BATCH SUSPEND and BATCH RESUME commands). See page 120 for information on the BATCH SUSPEND command. See page 111 for information on the BATCH RESUME command.

  - Canceling jobs (BATCH CANCEL command). See page 85.

  - Forcing ERROR job control option processing to stop, or forcing a running job to be canceled and discarded without normal completion and ERROR option processing (BATCH FORCE command). See page 99.

  - Holding jobs (BATCH HOLD command). See page 102.

  - Releasing jobs from HOLD status (BATCH FREE command). See page 101.

- Running a specified job on command (BATCH START command). See page 112.

- Changing job control options (BATCH CHJOB command). See page 88.

- Changing and/or confirming alternate user IDs (BATCH ALTID command). See page 83.

- Controlling task virtual machines:

  - Getting status information on task virtual machines (BATCH QUERY TASK command). See the section on BATCH QUERY, *VM Batch Facility User's Guide*.

  - Suspending and resuming execution in a task machine (BATCH SUSPEND and BATCH RESUME commands). See page 120 for information on the BATCH SUSPEND command. See page 111 for information on the BATCH RESUME command.

  - Halting execution and putting task machines in HOLD status (BATCH DRAIN and BATCH FORCE HOLD commands): See pages 97 and 99.

    - While jobs are running, canceling the jobs.
    - After task machines finish current jobs.

  - Canceling a job and freeing a task machine to run other jobs (BATCH CANCEL command). See page 85.

  - Forcing off a task machine (BATCH FORCE command). See page 99.

  - Restarting task virtual machines from HOLD or OFFLINE status (BATCH START command). See page 112.

  - Having a task machine run a specified job on command (BATCH START command). See page 112.

  - Scheduling LLS activity (BATCH LLS command). See page 106.

Starting the VM Batch Facility Control Program

To start the VM Batch Facility, you must start the VM Batch Facility control program in the monitor virtual machine. If you have not set up the VM Batch Facility for autologging, you will need to restart it after each system IPL (and whenever you stop the VM Batch Facility, as well).

After the first time you start the VM Batch Facility, it will usually do a *warm* start. That is, it will return to the state it was in when it stopped. It will restart any running jobs that were interrupted by a system shutdown. (A running job is restarted unless the RESTART NO option was in effect.) The VM Batch Facility will also restore queued jobs to their status before the shutdown.

Use the following procedure to start the control program.

Starting the VM Batch Facility control program

1. Log on to the VM Batch Facility monitor virtual machine.

2. After the virtual machine has finished IPLing CMS, the PROFILE EXEC will execute and start the control program by executing RUNBATCH EXEC.

   You will see messages showing that the VM Batch Facility is starting, showing the CONTROL FILE entries that will be in effect, and showing whether the VM Batch Facility has successfully started.

3. Enter the following command to put the monitor machine into disconnected mode:

   ```
   #cp disc
   ```

   The monitor machine should be in disconnected mode when the VM Batch Facility is running.

Usage Notes

1. **Cold starting the VM Batch Facility:** If necessary, you can have the VM Batch Facility perform a *cold* start by erasing the BATCH USERS, BATCH JOBS, BATCH SYSTEM and BATCH MACHINES files. These files are located on the monitor machine 194 minidisk. Cold starting may be required, for example, if you have received an error message warning of a problem with one of the BATCH files, and you have been unable to correct the problem by following the user response instructions in the error message documentation.

   Cautions: Cold starting the VM Batch Facility makes it impossible to start any queued jobs. Therefore, any queued jobs should be erased from the monitor machine 193 minidisk, and the monitor punch should be purged before cold starting the VM Batch Facility.

   Job owners must resubmit the erased jobs in order to have them run.

   Also, be sure that no jobs are running when the cold start takes place; no job completion processing can be done for these jobs when they finish running.

2. **RUNBATCH Return Codes**

   The return codes from RUNBATCH EXEC (and DGRMAIN MODULE, which is invoked by RUNBATCH) use the following convention:

Return Code	Meaning
0	The monitor has successfully shutdown in response to a STOP command. All running jobs have completed.
4	The monitor has successfully shutdown in response to a STOP NOW command, and at least one job was left running.
8	The monitor has successfully shutdown in response to a STOP JOBS command and at least one job was cancelled.
12	An error occurred during the monitor initialization. The monitor failed to start.
16	A condition has arisen preventing the continued operation of the monitor. The running monitor has shut itself down.

Enabling Users to Use the VM Batch Facility

There are different procedures for enabling each of three types of users to send jobs and commands to the VM Batch Facility. The three types of users are:

1. General users–users who can submit and control their own jobs using the commands described in the *VM Batch Facility User's Guide*.

2. Authorized users–users who can enter all VM Batch Facility commands, and who can operate, administer, and customize the VM Batch Facility.

3. General users at remote nodes or remote RJE (Remote Job Entry) workstations. See the *VM Batch Facility User's Guide* if you need more information on using the VM Batch Facility from remote nodes and RJE workstations.

The following are procedures for adding each type of user to the VM Batch Facility:

Adding General Users

1. Make sure that the user has links to the MAINT 19D and 19E minidisks. VM Batch Facility online help files and user interface files are stored on these system minidisks.

2. Make sure that the user has a link to the ISPF minidisk if the user is to use the VM Batch Facility panel interface. The required link and access commands can be added to each user's PROFILE EXEC or to the BATCH EXEC.

3. If the Resource Access Control Facility (RACF) is installed in your system and the use of alternate userids is required, ensure the required RACF PERMITs have been issued.' See the *RACF Security Administrator's Guide* for information about allowing the use of alternate user IDs.

The user can now enter nonprivileged commands to submit and control his or her own jobs.

┌─── **Adding authorized users** ─────────────────────────────────┐

1. Ensure that the user has links to the MAINT 19D and 19E minidisks. VM
   Batch Facility online help files and user interface files are stored on these
   system minidisks.

2. Identify the user to the VM Batch Facility by adding an ADMIN entry to the
   CONTROL FILE. For *user ID*, enter the user ID of the authorized user. For
   node ID enter the node ID if the authorized user is on a remote node. Be
   sure to use capitals for any letters in the user ID and node ID.

   ADMIN *USERID* [*NODEID*]

   This user will be able to enter any VM Batch Facility privileged command.
   You may identify a maximum of 16 authorized users. Authorized users
   need only the standard CP class G command authority.

3. Tailor the authority over the VM Batch Facility that the user will have:

   - If the user should be the default console message recipient, record the
     authorized user's user ID in the CONSOLE keyword. If the authorized
     user is on a remote node, their node ID is also required.

     CONSOLE *USERID* [*NODEID*]

   - If the user should be the default alert message recipient, make sure that
     the user's ADMIN keyword is the first ADMIN keyword in the CONTROL
     FILE. The first authorized user listed in the file is the default alert
     message receiver. This user ID may not be on a remote node.

   - If the user will be allowed to customize the VM Batch Facility, let the
     user have WRITE access to the monitor 199 minidisk. The CONTROL
     FILE is stored on the 199 minidisk. A user who can edit that file can
     customize the VM Batch Facility monitor.

4. Enter the BATCH CONTROL SET command or stop and restart the VM
   Batch Facility to put the changes you have made into effect. When
   CONTROL FILE processing has completed, the authorized user can
   operate, administer, and optionally customize the VM Batch Facility.

└───┘

┌─── **Adding Users at Remote Nodes and RJE Workstations** ───────┐

1. Give the user ID of the monitor virtual machine to the user. Commands
   from RJE workstations must be sent directly to the monitor virtual machine.

2. Change the VM Batch Facility user exits to let RJE and remote node users
   submit jobs, enter commands, and receive messages. See Chapter 5, "VM
   Batch Facility Installation Exits" on page 121 for help in changing the exits.
   This step is required only the first time you enable an RJE or remote node
   user to make use of the VM Batch Facility.

└───┘

Usage Note

Reserved user IDs and node IDs: Users whose user IDs or node IDs match certain VM Batch Facility command keywords should not be permitted to use the VM Batch Facility. See page 5 for a list of reserved user and node IDs.

Changing Monitor Machine Passwords

If your installation changes virtual machine passwords periodically, follow the directions shown in the following box to change monitor machine passwords.

How to Change Monitor Machine Passwords

1. Enter the BATCH DRAIN ALL or BATCH STOP command to stop processing of jobs.

2. Change the password of the monitor machine.

3. Log on to the monitor machine and restart the control program (if necessary) with this command:

   ```
   runbatch
   ```

4. Disconnect the monitor machine by entering:

   ```
   #cp disc
   ```

Changing Task Virtual Machine Passwords

If your installation changes virtual machine passwords periodically, follow the directions shown in the following box to change task machine passwords.

How to Change Task Machine Passwords

1. Before you change the password of a task machine, enter this command to temporarily stop the task machine:

   `batch drain userid`

   The command will place the task machine in HOLD status when it finishes the job it is running, if any.

2. Change the passwords of the task machine.

3. Edit the CONTROL FILE to record the new passwords in the TASK keyword that identifies the task virtual machine. Refer to the *Program Directory* for detailed instructions on how to update the CONTROL FILE.

   Be sure that all passwords are the same: the new Resource Access Control Facility (RACF) password (if RACF is running in your system), the password you put in the CONTROL FILE, and the password you put in the CP directory entry.

4. To put the new passwords into effect enter:

   `batch control set`

5. Restart the task virtual machine by entering:

   `batch start userid`

Using this procedure ensures that the VM Batch Facility will not try to AUTOLOG a task machine when it does not have the correct password. If it does try to AUTOLOG a task machine with an incorrect password, you will have to enter the START command to make the task machine available to the VM Batch Facility again. The VM Batch Facility will mark the task machine OFFLINE, and will not try to start it again after an AUTOLOG attempt fails.

Releasing Jobs from HOLD Status

A job is placed in HOLD status by the VM Batch Facility when there is no task machine that can run it.

This may be because:

- No task machine has been configured that will accept jobs for the class under which the job was submitted.

- No task machine has a start window that is compatible with the start window specified (or defaulted) for the job.

If the presence of a job with status HOLD is not intentional, the most likely causes are:

- The TASK and CLASS statements in the CONTROL file do not match. It may be that

  - A class has been defined without at least one task machine that will accept it.

  - The start windows of all task machines that accept a particular class do not overlap with the start window defined for the class.

    Any users who submit jobs under that class and who let the start window of the job default to that provided by the class, will find that their jobs are being placed in HOLD status.

- A user has submitted a job and has specified an override of the default start window provided for the class; and the override is not compatible with any task machines capable of accepting jobs of the class.

Releasing Jobs from HOLD

A job can be released from HOLD status by:

1. Using the BATCH CHJOB command to change the class or the start window of the job to a value that is compatible with configured task machines.

2. Re-configuring the relationships between the CLASSes and TASKs defined in the CONTROL FILE. This may be done while the monitor is running, by changing the CONTROL FILE and using the BATCH CONTROL SET command to put the changes into effect.

When an action resolves the condition causing the HOLD status on a job, the job is automatically released by the VM Batch Facility

A job in hold status can also be forcibly started using the task specific form of the BATCH START command, effectively overriding any class or window constraints on a task machine.

Using VM Batch Facility Job Accounting Facilities

If you have activated the VM Batch Facility Job Accounting exit (DGRUAC EXEC) by setting the ACCOUNT keyword in the CONTROL FILE as 1 or 3, then you must maintain any output from that exit according to your accounting requirements. The monitor does not tolerate a malfunction of the DGRUAC EXEC. If the exit issues a non-zero return code, then the VM Batch Facility will shut down.

In particular, if you have chosen to implement DGRUAC EXEC as it was provided with the VM Batch Facility, then you must regularly review the accounting file (DGRUAC RECORDS) created by the exit. As supplied with the product, the DGRUAC EXEC functions as follows:

- Accounting records are stored on the monitor 195 minidisk. The records are stored in the DGRUAC RECORDS file, the only file on the minidisk. See "Job Accounting Exit (DGRUAC EXEC)" on page 135 for a sample DGRUAC RECORDS file record.

- The Facility alertee will receive warning messages from the accounting exit when the monitor 195 minidisk becomes 85% and 90% full.

- The exit erases the DGRUAC RECORDS file from the monitor 195 minidisk if the minidisk becomes 95% full. The file is sent to the Facility alertee before it is erased to ensure that data is not lost.

VM Batch Facility Command Return Codes

The return codes from all VM Batch Facility commands use the following convention:

Return Code	Meaning
0	Command has been issued successfully.
4	Command has been issued, but there were complications; check for other messages.
8	Command failed due to parameter error.
12	Command not recognized or not permitted.
16	Command failed due to environmental error.

Note: The return codes to VM Batch Facility commands only indicate the success of transferring the command to the monitor. They give no indication of the monitor's response to the command.

BATCH ALERT: Controlling Alert Messages

Use the BATCH ALERT command to:

- Find out which user ID currently receives VM Batch Facility alert messages.
- Direct alert messages to a new user ID.

Alert messages are automatically sent out by the VM Batch Facility to warn of unusual conditions, such as stalled task machines.

```
┌─ Format ─────────────────────────────────────────────┐
│                                                        │
│ ►►──BATCH ALERT──────────►◄                            │
│                  └─userid─┘                            │
│                                                        │
└────────────────────────────────────────────────────────┘
```

no option

Enter BATCH ALERT without any option to find out which user ID in your system is currently the designated receiver of alert messages.

userid

Enter the user ID that should receive alert messages. The user ID will receive all alert messages issued by the VM Batch Facility after the BATCH ALERT command is processed.

The user you designate:

1. Must be logged on and receiving warning messages when you enter the command.

2. Can be any user in the system in which the VM Batch Facility is running except a VM Batch Facility task machine.

3. Should remain logged on or make arrangements to record messages so that alert messages are not lost.

You can designate only one user at a time as the alert message receiver.

```
┌─ Examples ───────────────────────────────────────────┐
│                                                        │
│ Enter this command to find out which user ID is receiving VM Batch Facility │
│ alert messages:                                        │
│                                                        │
│ batch alert                                            │
│                                                        │
│ The response from the command is a message naming the alert message │
│ receiver.                                              │
│                                                        │
│ Direct alert messages to a new user ID–CHARLIE—with this command: │
│                                                        │
│ batch alert charlie                                    │
│                                                        │
│ Both you and user ID CHARLIE will receive messages confirming the change. │
│                                                        │
└────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Verifying that a command has been completed:** When you change the designated alert message receiver, the VM Batch Facility will confirm the change with a message at your terminal. The VM Batch Facility also notifies the new message receiver.

2. **Receiver reset:** Changes made with the BATCH ALERT command remain in effect only until the VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is entered. When either of those events occurs, the alert message receiver is reset to the first authorized user identified in the CONTROL FILE.

BATCH ALTID: Running Jobs on an Alternate User ID

Use the BATCH ALTID command to confirm, or set and confirm, an alternate user ID for a job.

When job submittors request an alternate user ID for a job that is not the same as their own user ID, or when a job from a remote node is received, then the requested alternate user ID for the job must be confirmed by an administrator before the monitor will accept it. This is done with the BATCH ALTID command. Under normal circumstances it is preferable to have the DGRUJB user exit enter this command when required. See "Enabling a Facility for Remote Processing" on page 127 for more information.

```
  ┌─ Format ──────────────────────────────────────────────────┐
  │                                                            │
  │  ►►──BATCH ALTID──┬─userid─┬──┬─nodeid─┬──jobid──altid──►◄  │
  │                   └──*─────┘  └──*─────┘                    │
  │                                                            │
  └────────────────────────────────────────────────────────────┘
```

userid
> Enter the user ID of the job owner. You can specify an asterisk (*) to refer to your own user ID.

nodeid
> Enter node ID from which the job was submitted. You can specify an asterisk (*) to refer to the node that you are logged on to.

jobid
> Enter the job ID.

altid
> Enter the alternate user ID that the VM Batch Facility is to use when starting the job specified by *userid nodeid jobid*.

```
  ┌─ Example ──────────────────────────────────────────────────┐
  │                                                            │
  │  Enter this command to run CHARLIE's job TOOMUCH under the  │
  │  alternate user ID SPARE1:                                  │
  │                                                            │
  │  batch altid charlie * toomuch spare1                      │
  │                                                            │
  │  An asterisk (*) is specified for nodeid to indicate that   │
  │  you are logged on to the node that CHARLIE used to submit  │
  │  the job.                                                   │
  │                                                            │
  └────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are submitted to, but not accepted by the VM Batch Facility are not affected by this command.

2. **Jobs that have started:** Jobs that have already started may not have their alternate user ID changed except for recurring jobs, in which case the changed alternate user ID only takes effect on the next occurrence of the job.

3. **Job screening exit:** If the BATCH ALTID command is manually entered, it both sets and confirms an alternate user ID. It is then unnecessary for the exit to confirm the alternate user ID.

4. **Security Considerations:** In addition to confirming alternate user IDs before they are used, installations with an External Security Manager such as RACF may require that the owner of a user ID has permitted the facility to use it as an alternate.

5. **Alternate user ID for recurring jobs:** When an alternate user ID has been confirmed for a recurring job, it remains confirmed for all subsequent occurrences of the job.
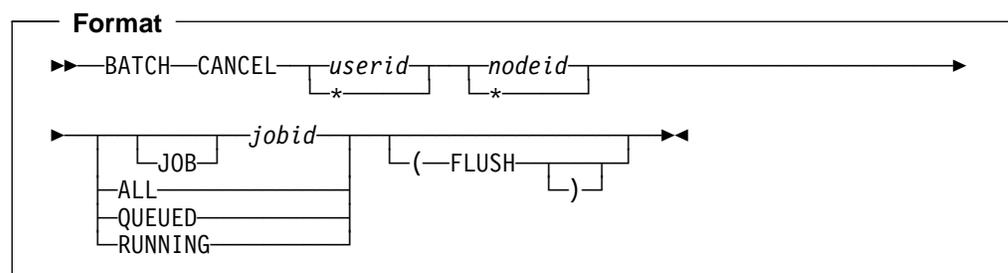
BATCH CANCEL: Canceling Jobs

Use the BATCH CANCEL command to cancel and discard one or more jobs that belong to a single user.

The effect of a cancel command on a job depends on the state of the job.

- If a job that is targeted by a cancel command does not recur:
    - If the job is running then the job is stopped. Unless the CONSOLE option for the job was specified as NO, then the job owner will receive the console log for the job up to that point. The processing that was specified by the ERROR job control option for the job then commences. On completion of the ERROR processing the job is discarded.
    - If the job has not yet run, then it is discarded immediately.

- If a job that is targeted by the cancel command recurs, then the cancel command can be used in two distinct ways. If the FLUSH option is not specified, then the cancellation only targets a running instance of the job. If the FLUSH option is specified, then the cancellation targets any current instance, as well as all future occurrences. So we may talk of cancelling a single instance of a recurring job, or of flushing the entire job.
    - If the job is running then the job is stopped and ERROR processing proceeds as for a non-recurring job. On completion of the ERROR processing, the job is discarded only if it was flushed; otherwise it is rescheduled to run again, either according to its recurrence pattern, or by having its chain condition re-imposed.
    - If the job is not running and was flushed, then the job is discarded immediately; otherwise the cancellation is ignored for the job.

BATCH CANCEL cannot be used to stop ERROR job control option processing that is being done for a canceled job. Use the BATCH FORCE command to stop ERROR option processing. The command is described in "BATCH FORCE: Logging Off Task Machines" on page 99.

```
 ┌─ Format ──────────────────────────────────────────────────────────────┐
 │                                                                        │
 │  ▶▶──BATCH──CANCEL──┬─userid─┬──┬─nodeid─┬────────────────────────▶    │
 │                     └─*──────┘  └─*──────┘                             │
 │                                                                        │
 │  ▶──┬──────────┬─jobid────┬──────────────┬──▶◀                         │
 │     │    ┌─JOB─┘          └─(─FLUSH──┬───┬┘                            │
 │     ├─ALL────────┘                   └─)─┘                             │
 │     ├─QUEUED─────                                                      │
 │     └─RUNNING────                                                      │
 │                                                                        │
 └────────────────────────────────────────────────────────────────────────┘
```

userid
   Specify the user ID of the owner of the jobs you want to cancel. You must specify a user ID to avoid an accidental cancellation of your own jobs.

   You can specify an asterisk (*) to refer to your own user ID.

nodeid
   Enter the node ID from which the job was submitted. You can specify an asterisk (*) to refer to the node that you are logged on to.

[JOB] *jobid*

To cancel a single job, specify the job ID of the job.

Use the following command to review job IDs if you are not sure of the one you want to cancel:

```
batch status userid all
```

You must precede the job ID with JOB if the job ID matches any BATCH CANCEL command option:

JOB
ALL
QUEUED
RUNNING

ALL

To cancel all of a user's jobs, enter the ALL option. The command will cancel:

- All running and suspended jobs. These jobs will be interrupted and will not be restarted.
- All queued jobs, including jobs in HOLD status.

The command will not cancel CLASS AUTOLOG jobs that have already started to run.

QUEUED

Enter this option to cancel all of a user's queued jobs. The command will cancel all queued jobs, including jobs in HOLD status. This option will only affect recurring jobs when the flush option is also specified.

RUNNING

Enter this option to cancel all of a user's running jobs. The command will cancel running and suspended jobs. The command will not cancel CLASS AUTOLOG jobs that have already started to run.

FLUSH

Enter this option to cancel any executing instance and all future occurrences of recurring jobs. If not specified, then only executing instances of recurring jobs will be cancelled.

Examples

Enter this command to cancel the job with the job ID NOGOOD, submitted by user CHARLIE on this node:

```
batch cancel charlie * nogood
```

The command shown below would cancel all jobs submitted by user CHARLIE on this node:

```
batch cancel charlie * all
```

Usage Notes

1. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are submitted to, but not accepted by the VM Batch Facility are not affected by this command.

2. **As a spawned command:** BATCH CANCEL can be issued from within a VM BATCH Facility job to cancel a spawned job with the same ownership, but the general user format described in *VM Batch Facility User's Guide* must be used.

3. **Confirmation Message:** For each job that is cancelled as a result of the command, you and the job owner will receive a message indicating that the cancellation process has been initiated. If the job was not running, the message also means that the cancellation is complete; but if the job was running then cancellation completes only when the actions that were specified for the ERROR option on the job submission have been performed. Until then, jobs from the same user with the same job ID will be rejected as duplicates. The time taken to complete the cancellation process is dependent upon the nature of the processing invoked for the ERROR option. An ERROR specification of NODUMP significantly reduces the duration of cancel processing.

4. **Job console log:** Unless the CONSOLE option for the job was specified as NO, the VM Batch Facility sends the job console log to the job owner if you cancel a running job. In all cases, the job owner receives a message if you cancel a job.

5. **Cancellation of chained jobs:** Unless it has already started to run, a chained job is canceled if the job to which it is chained is canceled. If the chain is one of recurring jobs, flushing a job will also flush all other jobs with a chain dependency on the flushed job. However, chained jobs that are running will be allowed to complete.

6. **Suspended task machines:** Canceling a suspended job frees the suspended task machine to run other jobs. The task machine status will return to READY as soon as ERROR option processing for the canceled job is completed.

7. **General user BATCH CANCEL command:** You can also use the nonprivileged form of the BATCH CANCEL command to cancel your own jobs. The nonprivileged command is identical with the authorized user command except that *userid* and *nodeid* are not included in the command.

BATCH CHJOB: Changing Job Control Options

Use the BATCH CHJOB command to change the job control options for submitted jobs.

```
┌─ Format ─────────────────────────────────────────────────────────────────┐
│  ►►──BATCH CHJob──┬──userid──┬──┬──nodeid──┬──jobid──(──┤ Options ├──┬─────┬──►◄   │
│                   └──*───────┘  └──*────────┘            └──)──┘      │
│                                                                            │
│  Options:                                                                  │
│  ├──┬──────────────────────┬──┬─────────────────────┬──────────────────►   │
│     └─ACcount──account──┘     └─ALtid──altuserid─┘                          │
│                                                                            │
│  ►──┬──────────────────────────────────────────────────────┬──┬──────────────────────┬──►   │
│     └─CHain─┬──┬──userid──┬──┬──.nodeid──┬──.jobid──┬──────────────┬──┬──┘  └─CLass──classid─┘   │
│            │  └──*───────┘  └──.*────────┘          └──.jobstep──┘  │                         │
│            └─No────────────────────────────────────────────────────┘                         │
│                                                                            │
│  ►──┬───────────────────────┬──┬──────────────────┬──┬──────────────────────┬──►   │
│     └─CONSole─┬─Yes───┬──┘    └─DAte─┬─mm/dd/yy──────┬──┘  └─ERror─┬─DUmp─────┬──┘   │
│              ├─No─────┤             ├─mm/dd/ccyy────┤             ├─NODump───┤        │
│              ├─Rc─────┤             ├─dd.mm.yy──────┤             └─filename─┘        │
│              └─Debug──┘             ├─dd.mm.ccyy────┤                                 │
│                                    ├─HOLD──────────┤                                 │
│                                    └─OPEN──────────┘                                 │
│                                                                            │
│  ►──┬─LLS───┬──┬──────────────────────────┬──┬─PJ──n─┬──┬─PRint─┬─pppp─┬──┬──►   │
│     └─NOLLS─┘  └─PArms⁽²⁾─┬─(──▼─parms──)──┬──┘       └──────┘  └──────┘  └─NONE─┘   │
│                         └─()──────────────┘                                 │
│                                                                            │
│  ►──┬─PUnch─┬─uuuu─┬──┬──┬─REPeat─┬─ffffffff─┬──┬──┬─REstart─┬─Yes─┬──┬──►   │
│     └───────└─NONE─┘  │  └────────├─HOLD─────┤  │  └─────────└─No──┘  │   │
│                                  ├─NOHOLD────┤                               │
│                                  ├─NEVER─────┤                               │
│                                  └─KEEP──────┘                               │
│                                                                            │
│  ►──┬─SEconds─┬─sssssss──┬──┬──┬─STORage─┬─kkkkkkkK─┬──┬──┬─WBegin─┬─hhmm─┬──┬──►   │
│     └─────────└─NONE─────┘  │  └─────────├─kkkkkkk──┤  │  └────────└─OPEN─┘  │   │
│                                        └─mmmmM─────┘                         │
│                                                                            │
│  ►──┬─WEnd─┬─hhmm─┬──┬──┤                                                    │
│     └──────└─OPEN─┘  │                                                       │
│                                                                            │
│  Notes:                                                                     │
│  ¹ You can enter options in any order after the open parenthesis.           │
│  ² Parameters inside the parentheses including blanks, cannot exceed a total of 18 characters in   │
│    length.                                                                  │
└────────────────────────────────────────────────────────────────────────────┘
```

userid

Enter the user ID that submitted the job. You can specify an asterisk (*) to refer to your own user ID.

nodeid

Enter the node ID from which the job was submitted. You can specify an asterisk (*) to refer to the node that you are logged on to.

jobid

Enter the job ID whose control options you want to change.

If you are not sure of the job ID, use the BATCH STATUS command to review the job IDs of submitted jobs.

For the details of the job control options, see the section 'Using the Job Control Options of the SUBMIT and CHJOB commands' in the *VM Batch Facility User's Guide*.

The following table suggests ways to use the BATCH CHJOB command options.

Table 1 (Page 1 of 2). Changing Job Execution with the BATCH CHJOB Command.

To make this change:	Use this option:	When changes permitted:*
Charge a different account number.	ACcount	Only before a job starts running.
Run job under a different alternate user ID.	ALtid	Only before a job starts running.
Make job start dependent on the status of another job, or change or cancel an existing job chain.	CHain	Only before a job starts running.
Assign the job to a different class.	CLass	Only before a job starts running.
Change the disposition of a job's console log.	CONSole	Only before a job starts running.
Change the start date.	DAte	Only before a job starts running.
Place a job on indefinite hold.	DAte	Only before a job starts running.
Remove start date restrictions.	DAte	Any time after you submit a job.
Change the action to take if a job is canceled.	ERror	Any time before a job ends.
Subject job to LLS activity (if permitted).	LLS	Any time after the job is submitted.
Do not subject job to LLS activity (if permitted).	NOLLS	Any time after the job is submitted.
Change or cancel the parameters to be passed to a job.	PArms	Only before a job starts running.
Change the logon password of a CLASS AUTOLOG job.	PASSword	Only before a job starts running.
Change job priority relative to your other queued jobs.	PJ	Only before a job starts running.
Change the recurrence pattern for a job.	REPeat	At any time.
Specify whether a job should be restarted if it is interrupted by a system shutdown.	REstart	Any time before a job ends.
Change resource use limits.	SEconds PRint PUnch	Any time before a job ends.
Change the virtual storage size of a job	STORage	Only before a job starts running.

BATCH CHJOB Command

Table 1 (Page 2 of 2). Changing Job Execution with the BATCH CHJOB Command.

To make this change:	Use this option:	When changes permitted:*
Change the start time window, the period of time during which a job may start.	WBEGIN WEND	Only before a job starts running.

Note: * Recurring jobs may be changed at any time. If an instance of the job is running when the CHJOB is issued, this column provides an indication of whether the change will apply to the currently running instance, or whether it will only take effect the next time the job runs.

Usage Notes

1. **As a spawned command:** BATCH CHJOB can be issued from within a VM BATCH Facility job.

2. **Option values:** The values of the job options and parameters must **not** contain non-displayable characters (e.g. x'00', x'15' etc...). Any non-displayable characters present will be translated to spaces (x'40').

3. **Options unchanged:** Options that you do not use in the BATCH CHJOB command stay set to their current values.

4. **Changing Job Class:** If you change the job class, the default LLS, WBEGIN, WEND, SECONDS, PRINT, PUNCH and STORAGE values of the job class are put into effect for your job. To prevent these options from being changed, specify values for them when you change the job class.

5. **Class Autolog Jobs:** The BATCH CHJOB command cannot be used to change the job class of a CLASS AUTOLOG job or to switch a non AUTOLOG job to class AUTOLOG. In such a case, cancel the job and resubmit it with the new class.

6. **Changing to Recurring Jobs:** The BATCH CHJOB command cannot be used to change a non-recurring job to a recurring job.

7. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are submitted to, but not accepted by the VM Batch Facility are not affected by this command.

8. **Options not applicable:** The following BATCH SUBMIT options are not applicable to BATCH CHJOB:

   - APpend option
   - ID option
   - WAIT option

9. **Changing job resource limits:** VM Batch Facility administrators may change the limits (seconds, print, and punch) imposed for a job regardless of the job's CLASS definition.

BATCH CONSOLE: Controlling Console Message Recording

Use the BATCH CONSOLE command to control logging of console messages issued by the VM Batch Facility monitor machine. The VM Batch Facility logs console messages by recording them in a spool file.

With BATCH CONSOLE, you can:

- Find out which user ID in your system is currently the designated receiver of the console message log file.
- Direct the console message log file to a new message receiver.
- Start recording console messages.
- Stop recording console messages.
- Close the console message log file and send it to the designated receiver of console messages.

```
┌─ Format ──────────────────────────────────────────────────────┐
│                                                                │
│  ►►──BATCH CONSOLE──┬──────────────────────────────┬──►◄       │
│                     │  ┌─userid─┬──────────┬        │          │
│                     │           ├─nodeid─┤           │          │
│                     │           └─*──────┘           │          │
│                     ├─START──────────────────────────┤          │
│                     ├─STOP───────────────────────────┤          │
│                     └─CLOSE──────────────────────────┘          │
│                                                                │
└────────────────────────────────────────────────────────────────┘
```
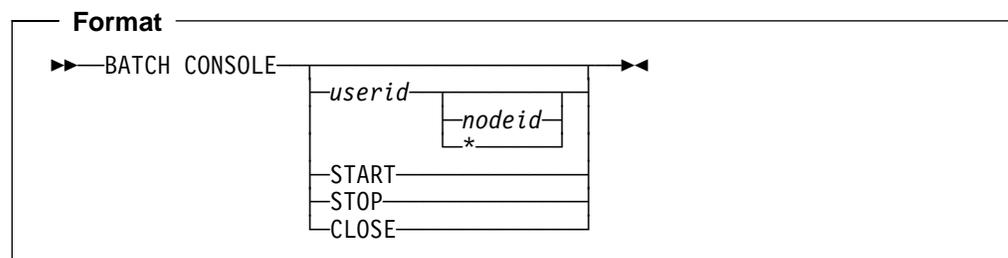
No option

> To find out the user ID of the current console message receiver, enter BATCH CONSOLE without any option. The VM Batch Facility will display a message at your terminal with the user ID of the message receiver.

userid

> To have console messages spooled to a new message receiver, enter the user ID of the new receiver. You can designate only one user at a time as the console message receiver.

> You can direct console messages to any user ID in your system except for VM Batch Facility task machines. However, only VM Batch Facility authorized users can enter the BATCH CONSOLE CLOSE command to review the console log file.

> All new console messages will be spooled to the user ID you specify. The entire contents of the current console message log file will also be spooled to the user ID you specify.

nodeid

> Enter the node ID where the job was submitted. You can specify an asterisk (*) to refer to the node that you are logged on to.

START

> Enter this option to have the VM Batch Facility start recording console messages.

STOP

> Enter this option to have the VM Batch Facility stop recording console messages. The VM Batch Facility will not keep any records of its activity while console spooling is stopped. Stopping console spooling does not erase any

existing console log file, nor does it send the console log file to the console message receiver.

BATCH CONSOLE STOP does not stop the VM Batch Facility from sending out alert messages.

CLOSE

Enter this option to make the console message log file available for review. This option directs the VM Batch Facility to close the spool file and send it to the reader of the console message receiver. The VM Batch Facility will start a new console message log file as soon as the old log file is closed, if console spooling is set to start.

Examples

To start console message spooling, enter:

```
batch console start
```

The VM Batch Facility will start recording BATCH monitor machine console messages in a spool file. The file is spooled to the user ID designated to receive the log file.

To direct the console message log file to user CHARLIE, enter:

```
batch console charlie
```

The current log file will be spooled to CHARLIE.

To stop console message spooling, enter:

```
batch console stop
```

No more console messages will be recorded. If console message spooling was active before this command, the log file will remain spooled to the designated user ID. To close the file and make it available to the designated user, enter:

```
batch console close
```

Usage Notes

1. **Verifying that the command has been completed:** The VM Batch Facility confirms that your command has been executed with a message displayed at your terminal.

2. **Background information:** Console message recording keeps a record of VM Batch Facility activity to help you audit the batch processing done in your system. You control recording of console messages with the CONSOLE keyword and the BATCH CONSOLE command.

   If you activate console message recording, the VM Batch Facility records console messages in a spool file, the console message log file. The file is spooled to a user whom you designate as the console message receiver. The file is available to the receiver only when you use the BATCH CONSOLE CLOSE command to close the file and send it to the receiver.

3. **Types of information logged:** VM Batch Facility console messages record the activity of the monitor machine. Included in the console log file are

messages that echo all commands received by the VM Batch Facility, as well as responses to commands, error messages, and alert messages.

4. **Preventing the log file from becoming too large:** Be sure to CLOSE the spool file and process or discard it at frequent intervals. The file uses system spool space and can become quite large.

5. **Relation to CONSOLE keyword:** Use the CONSOLE customizing keyword to set the default status of console message logging. You can use the CONSOLE keyword to specify a default message receiver or to specify that no logging be done until a BATCH CONSOLE START command is entered.

6. **Console spooling reset:** Changes made with the BATCH CONSOLE command remain in effect only until the VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is entered. When either of those events occurs, console message logging is reset to the status recorded in the CONSOLE keyword in the CONTROL FILE.

7. **Job console log:** Jobs generate a separate console log file when they are run. The job console log file records console messages issued by the task machine running the job. Job log files are automatically closed and sent to the job owner when a job is completed or canceled.

BATCH CONTROL: Putting a New Control File into Effect

Use the BATCH CONTROL command to:

- Review the current keyword values in the CONTROL FILE.

- Put your changes to the CONTROL FILE into effect (make customization changes that you have made in the CONTROL FILE take effect in the VM Batch Facility).

Format

```
►►──BATCH CONTrol────────────►◄
                 └─SET─┘
```

---

no option

Enter BATCH CONTROL without any options to review the current CONTROL FILE keyword values. The VM Batch Facility will display the values as a series of messages at your terminal.

The command will not display TASK and CLASS entries in the CONTROL FILE. Use the BATCH QUERY TASK ALL and BATCH QUERY CLASS ALL commands to review these entries.

Note: The keyword values you see may not match those in the CONTROL FILE on the monitor machine 199 minidisk. The messages show the values that are actually in effect in the VM Batch Facility. The messages will not reflect changes made to the CONTROL FILE that have not yet been put into effect or that have been rejected.

SET

Enter this option to put changes you have made to the CONTROL FILE into effect. The command directs the VM Batch Facility to copy the keyword values into storage and put them into effect.

Examples

Enter this command to review the CONTROL FILE entries that are currently in effect in the VM Batch Facility:

```
batch control
```

TASK and CLASS entries will not be included in the messages that will be displayed at your terminal. Use the BATCH QUERY TASK and BATCH QUERY CLASS commands to review task machines and job classes.

Enter this command to put a new CONTROL FILE into effect without stopping the VM Batch Facility:

```
batch control set
```

When the VM Batch Facility has finished processing the CONTROL FILE, you will see messages listing the CONTROL FILE entries that are now in effect, together with any error messages resulting from the CONTROL FILE processing.

Usage Notes

1. **Background:** See Chapter 3, "Customizing the VM Batch Facility" on page 19 for an explanation of the CONTROL FILE and how to use it to customize the VM Batch Facility.

2. **Verifying that changes are in effect:** The VM Batch Facility will display the new customizing values as soon as it finishes processing the CONTROL FILE.

3. **MAXIMUM-USERS and MAXIMUM-JOBS keywords:** These keywords cannot be changed with the BATCH CONTROL SET command. See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for help in changing the values of these keywords.

4. **Error handling:** If any required keyword is missing from the CONTROL FILE or has an invalid value, the VM Batch Facility uses the default value of the keyword. Figure 1 on page 20 shows the default values.

   If there are duplicate keyword entries where only a single entry is allowed, the VM Batch Facility uses the value of the first of the entries.

   Exceptions: the VM Batch Facility handles errors in TASK, CLASS, and ADMIN keywords as follows:

   - The VM Batch Facility ignores TASK keywords that have invalid values. The VM Batch Facility will not use the task machine identified by the invalid TASK keyword.

     If there are no valid TASK keywords, the VM Batch Facility will not have any task machines available to it.

   - The VM Batch Facility also ignores CLASS keywords that have invalid values.

     If there are no valid CLASS keywords, the VM Batch Facility will have only CLASS A or CLASS AUTOLOG available to it when CONTROL FILE processing is done. It reuses the current CLASS A if CLASS A was defined before the BATCH CONTROL SET command. It uses CLASS AUTOLOG if that was the only job class defined before the command was entered.

   - The VM Batch Facility ignores ADMIN keywords that are invalid. If there are no valid ADMIN keywords, the VM Batch Facility defines only the issuer of the BATCH CONTROL SET command as an authorized user.

   - If there are too many TASK, CLASS, or ADMIN keywords, the VM Batch Facility ignores the extra entries. The VM Batch Facility will use only the first 128 TASK entries, 256 CLASS entries, and 16 ADMIN entries.

   In most cases, the VM Batch Facility sends a warning message to the issuer of the BATCH CONTROL SET command if it finds any problem in the CONTROL FILE.

   Any missing or invalid keywords should be corrected before the VM Batch Facility is stopped and restarted. It may not be able to restart if there is an error in the CONTROL FILE.

5. **Effect of changes on queued jobs:** Certain changes in a new CONTROL FILE can prevent queued jobs from running. These changes include:

   - Changing or deleting task machines, leaving one or more job classes without task machines to run on.

- Changing task machine start time windows, so that job start windows do not overlap with the task machine windows.

Affected jobs are put in HOLD status and will not be started. Jobs can be released from HOLD status by correcting the CONTROL FILE or having the job owner change job control options.

See "Releasing Jobs from HOLD Status" on page 79 for help in avoiding errors and releasing jobs from HOLD status.

6. **Changes to job class values:** Jobs already queued when changes were made to a job class will be run under the job class values that were in effect when the jobs were submitted.

7. **Effect of changes on task machines:** If a task machine is deleted while it is running a job, the job will be allowed to end before the deletion is put into effect. Exception: the job will be canceled if there are 128 task machines defined after the deletion. In this case, the deletion of a task machine goes into effect immediately and any running job is canceled.

BATCH DRAIN: Stopping Task Machines from Running Jobs

Use the BATCH DRAIN command to temporarily stop one or more task machines from running jobs. BATCH DRAIN puts task machines into HOLD status. Task machines that are in HOLD remain logged off and unavailable for running jobs until they are released with the BATCH START command.

The BATCH DRAIN commands stop task machines only after any running jobs are finished.

```
┌─ Format ────────────────────────────────────────────────────────┐
│                                                                  │
│  ►►──BATCH DRAIN──┬─taskid─┬──►◄                                  │
│                   └─ALL────┘                                     │
│                                                                  │
└──────────────────────────────────────────────────────────────────┘
```

taskid
>   To drain a single task machine, enter the user ID of the task machine. If the task machine is running a job, the VM Batch Facility will wait until the job is done before putting the task machine into HOLD status. Otherwise, the task machine will go into HOLD status as soon as the command is received.

ALL
>   Use this option to drain all task machines. Task machines that are running will log off and go into HOLD status as they finish their jobs. All other task machines will be put into HOLD status as soon as the command is received.

```
┌─ Example ────────────────────────────────────────────────────────┐
│                                                                  │
│  To reduce the resource demand of the VM Batch Facility when the │
│  system work load is high, enter:                                │
│                                                                  │
│  batch drain all                                                 │
│                                                                  │
│  The command would shut down the task machines gracefully,       │
│  without interrupting or canceling jobs.                         │
│                                                                  │
└──────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **CLASS AUTOLOG jobs:** You cannot drain virtual machines being used to run CLASS AUTOLOG jobs. The VM Batch Facility will continue to start CLASS AUTOLOG jobs even after a BATCH DRAIN ALL command.

2. **Releasing task machines from HOLD:** Task machines can be released from HOLD status only with the BATCH START command. See page 112 for information on BATCH START.

3. **Verifying that the command is completed:** You will receive a task machine status message for each task machine affected by a BATCH DRAIN command.

4. **Canceling BATCH DRAIN commands:** Enter the BATCH START command to cancel a BATCH DRAIN command that has not yet gone into effect.

5. **Monitor virtual machine not affected:** The monitor virtual machine is not affected by BATCH DRAIN commands. It will continue to receive jobs and respond to commands.

6. **Suspended task machines:** The pending status of a suspended task machine will be changed to *Hold* when a BATCH DRAIN command is entered.

However, the task machine will not be put into HOLD status until the job has ended.

7. **ERROR job control option:** The BATCH DRAIN command does not prevent ERROR job control option processing from being done. If a job is canceled in a DRAINing task machine, ERROR processing will be done before the DRAIN goes into effect.

BATCH FORCE: Logging Off Task Machines

Use the BATCH FORCE command to log off a task machine, canceling the job that the task machine was running. You can use the BATCH FORCE command to:

- Log off a running task machine and make it available for running other jobs.

- Put the task machine into HOLD status after it is forced. (Task machines in HOLD status stay logged off and unavailable for running jobs until they are released with the BATCH START command.)

- Halt ERROR option processing and log off a task machine. BATCH FORCE and BATCH STOP JOBS are the only commands that can be used to halt ERROR option processing.

```
 ┌─── Format ──────────────────────────────────────────┐
 │                                                      │
 │  ►►──BATCH FORCE──┬─taskid──────┬──►◄                │
 │                   └─taskid HOLD─┘                    │
 │                                                      │
 └──────────────────────────────────────────────────────┘
```

taskid

Enter the user ID of the task machine that you would like to log off. The command will have these effects:

- The task machine is logged off right away.
- The job that was running is canceled and will not be restarted.
- No ERROR option processing is performed. Also, while the task machine's 191 minidisk is erased and other normal job completion processing is performed, files queued in the task's reader, printer, and punch are not purged until a new job is started.
- The task machine is available to run another job, provided that its start time window is still open.

taskid HOLD

To log off a task machine and put it in HOLD status, enter the user ID of the task machine and the HOLD option. The command will have these effects:

- The task machine is logged off as soon as the command is received.
- The job that was running is canceled and will not be restarted.
- No ERROR option processing is performed. Also, while the task machine's 191 minidisk is erased and other normal job completion processing is performed, files queued in the task's reader, printer, and punch are not purged until a new job is started.
- The task machine is placed in HOLD status. It will remain logged off and unavailable for running jobs until released with the BATCH START command. The task machine will be put in HOLD status whether it is running when the VM Batch Facility receives the command.

Examples

To force off a task machine (called BATCH1 in this example) and prevent it from running any other jobs, enter:

```
batch force batch1 hold
```

To force off BATCH1 and let it be used for other jobs (within any start time limitations), enter:

```
batch force batch1
```

In both cases, the job that was running in BATCH1 is canceled. No ERROR option processing is done. Files queued in the BATCH1 printer, reader, and punch are not purged.

Usage Notes

1. **Job console log:** A job's console message log file is sent to the job owner if a running job is forced and CONSOLE NO was not specified for the job.

2. **Accounting:** The accounting exit is called, (if it is in use), when a job is FORCEd. An accounting record is generated for the FORCEd job.

3. **Releasing task machines from HOLD:** Use the BATCH START command to release a task machine from HOLD status. For more information on BATCH START, see "BATCH START: Starting Jobs and Task Machines" on page 112.

4. **CLASS AUTOLOG jobs:** You cannot use BATCH FORCE to cancel CLASS AUTOLOG jobs.

BATCH FREE: Releasing Held Jobs

Use the BATCH FREE command to let a job that has been held, either by a
BATCH HOLD command, or by a exit, run.

```
┌─ Format ──────────────────────────────────────────────────────────┐
│                                                                    │
│  ►►──BATCH FREE──┬─userid─┬──┬─nodeid─┬──────────┬─jobid─┬──►◄      │
│                  └───*────┘  └───*────┘  ┌─JOB─┐            │
│                                          └─ALL─┘                   │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```

userid
>   Enter the user ID of the job owner. You can specify an asterisk (*) to refer to
>   your own user ID.

nodeid
>   Enter the node ID from which the job was submitted. You can specify an
>   asterisk (*) to refer to the node that you are logged on to.

jobid
>   Enter the job ID you want to release.
>
>   *jobid* must be prefixed with JOB to specify a job named ALL or JOB.

ALL
>   Specify ALL to release all *admin-held* and *exit-held* jobs queued for the user.

```
┌─ Example ─────────────────────────────────────────────────────────┐
│                                                                    │
│  To release user CHARLIE's job CHAJ01 after it had been held by an exit, while │
│  being logged on to the node that was used to submit the job, enter: │
│                                                                    │
│  batch free charlie * chaj01                                       │
│                                                                    │
│  To release all of user CHARLIE's jobs that have been held, enter: │
│                                                                    │
│  batch free charlie * all                                          │
│                                                                    │
└────────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Admin-held jobs:** Jobs that are *admin-held* can only be FREEd by an
   administrator, while jobs that are *exit-held* may be freed by the exit as well as
   administrator command.

2. **Preventing exits from holding jobs:** Jobs that have been freed by manually
   entering this command cannot subsequently be held by an exit.

   Therefore this command may be entered against a queued job, even though it
   is not yet held to prevent it from ever being held by an exit.

   See "Holding and Freeing Jobs by Exits" on page 129 for more information.

BATCH HOLD: Prevent Jobs from Being Dispatched

Use the BATCH HOLD command to prevent a queued job from being dispatched by the monitor.

Jobs that have been held by the BATCH HOLD command must be freed by BATCH FREE command before they will run.
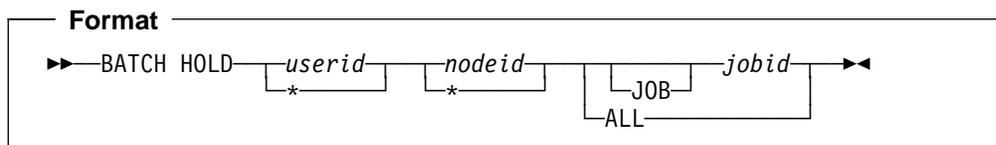
```
┌─ Format ─────────────────────────────────────────────────────────────┐
│                                                                       │
│  ►►──BATCH HOLD──┬─userid─┬──┬─nodeid─┬──┬──────┬──jobid──►◄          │
│                  └─*──────┘  └─*──────┘  ├─JOB──┤                     │
│                                          └─ALL──┘                     │
│                                                                       │
└───────────────────────────────────────────────────────────────────────┘
```

userid
   Enter the user ID of the job owner. You can specify an asterisk (*) to refer to your own user ID.

nodeid
   Enter the node ID from which the job was submitted. You may specify an asterisk (*) to refer to the node that you are logged on to.

jobid
   Enter the job ID you want to hold.

   jobid must be prefixed with JOB to specify a job named ALL or JOB.

ALL
   Specify ALL to prevent all jobs queued for the user from being dispatched.

```
┌─ Example ─────────────────────────────────────────────────────────────┐
│                                                                        │
│ To prevent user CHARLIE's job CHAJ01 from being dispatched, while being│
│ logged on to the node that was used to submit the job, enter:          │
│                                                                        │
│ batch hold charlie * chaj01                                            │
│                                                                        │
│ To prevent all of user CHARLIE's queued jobs from being dispatched, enter:│
│                                                                        │
│ batch hold charlie * all                                               │
│                                                                        │
└────────────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are submitted to, but not accepted by the VM Batch Facility are not affected by this command.

2. Jobs held by the BATCH HOLD command do not have status HOLD. If they were held by manually entering the BATCH HOLD command then their status is *Admin-held*. If they were held by return code from an exit, or by an exit entering the HOLD command, then their status is *Exit-held*.

3. Jobs can only be held before they start to run.

4. A job held by administrator command cannot be freed by an exit. A job held by an exit can be freed by an administrator command.

   See "Holding and Freeing Jobs by Exits" on page 129 for more information.

---

5. Jobs held may have their job control options changed by the BATCH CHJOB command, but may only be freed by the BATCH FREE command.

BATCH LEVEL: Reviewing Service Levels

Use the BATCH LEVEL command to interrogate the service levels of each serviceable component of the VM Batch Facility for problem diagnosis.

```
┌─ Format ──────────────────────────────────────────────────┐
│                                                            │
│   ►►──BATCH──LEVEL──►◄                                      │
│                                                            │
└────────────────────────────────────────────────────────────┘
```

BATCH LEVEL is primarily intended for use when service level information is requested by the IBM Support Center. Each component of the VM Batch Facility accessible from the user's virtual machine and the monitor is interrogated for its service level.

BATCH LEVEL will submit the DGRLEVEL EXEC on your behalf to the currently addressed monitor. DGRLEVEL EXEC obtains service level information for the VM Batch Facility components accessible by the task machine on which it runs.

DGRLEVEL EXEC requires a CLASS A task machine to run.

The output from BATCH LEVEL consists of several messages to the user's virtual console and the console log returned from the DGRLEVEL EXEC.

Usage Notes

1. **CP MSG must be ON.** Before issuing the command be sure to CP SET MSG ON, otherwise the results will be incomplete.

2. **Restrictions:** This command is not available via the RJE interface nor via the ISPF panel interface. It is not permitted to be issued from any facility installation exit nor from a task machine.

3. **Obtaining a record of the service level:** To obtain a record of the service level messages issued to your virtual console (e.g. to submit to IBM for problem determination) you must manually spool your console to yourself. An annotated sample console log follows. The commands you enter are shown in lower case, annotation lines begin with an asterisk (*).

┌─ **Example** ───┐

```
* Spool the console back to myself ...
cp spool cons start term to *
Ready;
* Issue the BATCH LEVEL command.  These service level messages
* originate from my virtual machine ...
batch level
DGRBAT555I BATCH Version 2.2.0, Service Level 000.
DGREXE120I YOU ARE NOW ADDRESSING MONITOR 'BATCH' AT NODE 'KOALA'
DGREXE555I DGRBATCH Version 2.2.0, Service Level 000.
DGRIPW555I DGRIPW Version 2.2.0, Service Level 000.
DGRUVM555I DGRUVM Version 2.2.0, Service Level 000.
DGRLAF555I DGRLAF Version 2.2.0, Service Level 000.
DGRCHJ555I DGRCHJO Version 2.2.0, Service Level 000.
DGRPAS555I DGRPASS Version 2.2.0, Service Level 000.
DGRSUB555I DGRSUBM Version 2.2.0, Service Level 000.
DGRUME555I DGRUME Version 2.2.0, Service Level 000.
DGREXE119I JOB 'DGRLEVEL' SUBMITTED TO 'BATCH' AT 'KOALA'
Ready;
* The next two messages originate from the monitor...
DGRMAI555I DGRMAIN Version 2.2.0, Service Level 000.
DGRRBT555I RUNBATCH Version 2.2.0, Service Level 000.
* This indicates that the DGRLEVEL EXEC has been submitted to
* the VM Batch Facility and it has run successfully ...
Job DGRLEVEL has been received
Job DGRLEVEL has started
RDR FILE 1222 SENT FROM REBECCA CON WAS 1944 RECS 0035
          CPY  001 T NOHOLD NOKEEP
Job DGRLEVEL has ended; RC = 0
* Now stop the console ...
cp spool cons stop close
RDR FILE 1220 SENT FROM REBECCA CON WAS 1220 RECS 0024
          CPY  001 C HOLD   NOKEEP
Ready;
```

└───┘

BATCH LLS: Scheduling LLS Activity

Use the BATCH LLS command to review or change the activity of the Load Level Scheduler of the VM Batch Facility.

The Load Level Scheduler is described in detail in the section "Load Level Scheduling" on page 64.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│                                                                   │
│  ►►──BATCH LLS─┬──────┬─┬────────┬─┬────────┬──►◄                 │
│               ├─CPU──┤ ├─llsscv─┤ ├─llsrcd─┤                      │
│               ├─STO──┤ └─*──────┘ └─*───────┘                     │
│               ├─PAG──┤                                            │
│               └─IO───┘                                            │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```
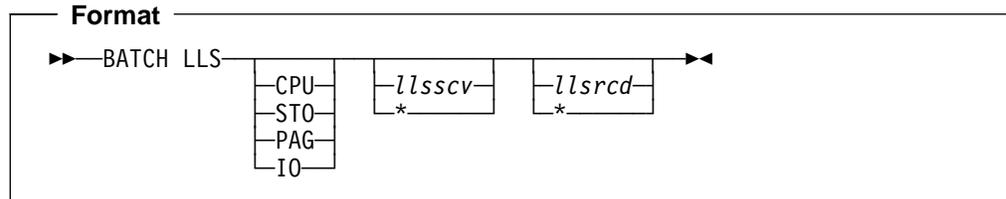
no option
> Enter BATCH LLS without options to display the current measured contention value, and the defined LLS threshold values, for all resources monitored by the LLS. The number of jobs that are suspended for contention of each resource is also displayed.

CPU , **STO** , **PAG** or **IO**
> Specify the system resource that is to have its LLS threshold values changed.

llsscv
> Specify the required Suspend Contention Value (SCV) for the resource. It must be a positive integer greater than zero.
>
> The VM Batch Facility will act to decrease its consumption of the resource if it detects a number of system users waiting for the resource greater than this number.

llsrcd
> Specify the required Resume Contention Decrement (RCD) for the resource. It must be a positive integer greater than zero and be smaller than *llsscv*.
>
> The RCD calculates the Resume Contention Value (RCV) for the resource as RCV = SCV - RCD. The VM Batch Facility will act to restore its consumption of the resource if it detects the number of system users waiting for the resource to be less than the RCV.

If either *llsscv* or *llsrcd* is specified as an asterisk (*), then LLS activity is deactivated for the selected resource.

```
┌─ Example ─────────────────────────────────────────────────────────┐
│                                                                   │
│  To schedule LLS activity for the system resource processor, with a SCV of 200 │
│  and a RCD of 20, enter:                                          │
│                                                                   │
│  batch lls cpu 200 20                                            │
│                                                                   │
│  To display the current values of the SCV, RCV and RCD for each resource, │
│  enter the BATCH LLS command without any options:                │
│                                                                   │
│  batch lls                                                        │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Default values:** The default values of the SCV and RCD for each resource can be defined in the CONTROL FILE using the LLS keyword. For more information, see "Filling Out the LLS Keyword" on page 69.

2. **Resume suspended jobs:** Jobs suspended by the LLS may be forcibly resumed with the BATCH RESUME command. For more information see "BATCH RESUME: Restarting SUSPENDed Jobs" on page 111.

3. **Job subjection to LLS:** Only jobs specified as subject to LLS activity (by virtue of their job class) are affected by the LLS.

BATCH PRIORITY: Setting Job Priority

Use the BATCH PRIORITY command to change the order in which jobs are run.

BATCH PRIORITY works by changing the priority of a job. The VM Batch Facility assigns the lowest priority to jobs as they are queued. By assigning a higher priority to important jobs, you can have those jobs run before jobs of lesser priority.

The BATCH PRIORITY command does not override start time and other limitations on a job. The command changes the priority of a job relative only to other jobs that are ready to run at the same time.

To run an urgent job right away, use the BATCH START command. For more information, see "BATCH START: Starting Jobs and Task Machines" on page 112.

```
┌─ Format ─────────────────────────────────────────────────────────┐
│                                                         ┌─0─┐     │
│  ►►──BATCH PRIority──┬─userid─┬──┬─nodeid─┬──jobid──┼───┼──►◄     │
│                      └─*──────┘  └─*──────┘          └─n─┘        │
└───────────────────────────────────────────────────────────────────┘
```
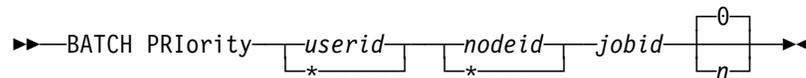
userid nodeid jobid

To assign the highest priority to a job, just enter the user ID of the job owner, the node ID from which the job was submitted and the job ID of the job. The command will assign the highest priority (0) to the job.

You may specify an asterisk (*) for *userid* to refer to your own user ID.

You may specify an asterisk (*) for *nodeid* to refer to the node that you are logged on to.

userid nodeid jobid n

To assign any priority to a job, enter the user ID and job ID that identify the job and a number from 0 to 9. The larger the number, the lower the priority. For example, a priority 9 job has lower priority than a priority 8 job. The default priority assigned to jobs as they are submitted is 9.

You may specify an asterisk (*) for *userid* to refer to your own user ID.

You may specify an asterisk (*) for *nodeid* to refer to the node that you are logged on to.

```
┌─ Example ─────────────────────────────────────────────────────────┐
│                                                                    │
│  To give user CHARLIE's job URGENT priority on the task machines   │
│  on which it can run, enter:                                       │
│                                                                    │
│  batch priority charlie * urgent                                   │
│                                                                    │
│  When its start time window opens, URGENT will be executed before  │
│  all other jobs that:                                              │
│                                                                    │
│    • Have lesser priorities.                                       │
│    • Can be run on the same task machine on which this task        │
│      machine is run.                                               │
└────────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are submitted to, but not accepted by the VM Batch Facility are not affected by this command.

   Note that in the case of a recurring job, this refers to the time when the job was last scheduled, rather than submitted.

2. **Equal priority:** If two jobs have the same start time and priority, the job that was submitted earlier will start first.

3. **Relation to job class priority on task machines:** Increased priority overrides the ranking of job classes on a task machine.

   For example, consider a task machine called BATCH1 that has this recorded in the class ID field of its TASK keyword entry: .ABCDEFG. If you raise the priority of a job belonging to class G, it will run before jobs of lesser priority that belong to job classes A, B, C, D, E, and F. Jobs of these classes normally have priority over class G jobs.

4. **Relation to PJ job control option:** The priority you set with BATCH PRIORITY is different from the priority set with the PJ job control option of BATCH SUBMIT and BATCH CHJOB. BATCH PRIORITY overrides any conflicting priority set with PJ. For example, if a job owner sets a higher priority on JOBA than on JOBB but you set a higher priority on JOBB, JOBB will run first. PJ option priority is taken into account only if a tie must be broken between jobs with equal priority that belong to the same user.

BATCH RELOAD: Refreshing Installation Exits

Use the BATCH RELOAD command to refresh copies of all running installation exits of a monitor. This command enables changes to installation exits to be effected without taking a monitor down.

A check is made to ensure that all installation exits are loaded successfully from the 199 disk before the old exits are dropped from storage. The command is rejected if any exit is not found on the 199 disk, and the monitor continues with the old exits.

```
┌─ Format ──────────────────────────────────────────────────┐
│  ►►──BATCH──RELoad──►◄                                      │
└────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Exits must be stored on the minidisk that the monitor has at address 199:** On installation, the exit execs must be stored on the minidisk linked by the monitor at address 199. Exits cannot be stored on any other minidisk. At startup, or on receipt of the BATCH RELOAD command, the VM BATCH Facility ensures that all installation monitor exits are successfully loaded from the 199 disk.

   At startup, if any monitor exit is missing, the monitor terminates. On receipt of BATCH RELOAD command, if any exit is missing, the monitor discards the command and reloads all the old exits.

2. **Identifying Exits:** A VM Batch Facility administrator can code a monitor exit to identify itself by displaying an identifying message on the monitor's console in response to a single parameter of '?'. This identifying message will appear on the monitor console at start time, and whenever the exits are dynamically refreshed.

BATCH RESUME: Restarting SUSPENDed Jobs

Use the BATCH RESUME command to restart a task machine and job that have been temporarily halted with the BATCH SUSPEND command, or have been suspended by the Load Level Scheduler.

When you enter the command, the task machine will restart the suspended job at the point at which it was halted.

```
┌─ Format ──────────────────────────────────────────────────────────┐
│  ►►──BATCH RESume──taskid──►◄                                      │
└───────────────────────────────────────────────────────────────────┘
```

taskid
  Enter the user ID of the task machine that you want to restart from suspension.

```
┌─ Example ─────────────────────────────────────────────────────────┐
│                                                                    │
│  To restart a suspended job in the BATCH1 virtual machine, you would enter: │
│                                                                    │
│  batch resume batch1                                               │
│                                                                    │
│  The command causes BATCH1 to resume running the suspended job at the │
│  point at which the job was halted.                                │
│                                                                    │
└───────────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Verifying that the command has been completed:** The VM Batch Facility displays a status message for the RESUMEd task machine after it completes the command.

2. **VMCF Usage:** A job that uses VMCF, steals the external new PSW, or that modifies the CMS environment in any way, is not supported and removes the SUSPEND and RESUME function for the duration of that job.

3. **LLS Suspension:** Jobs suspended by LLS are unconditionally resumed when the task machine is resumed. It also unconditionally prevents the job from suffering further LLS suspension, regardless of the job's Class definition. In the case of a recurring job, any original subjection to LLS is re-imposed on the next occurrence of the job.

BATCH START: Starting Jobs and Task Machines

Use the BATCH START command to do the following:

- Restart one or all task machines that have been put in HOLD or OFFLINE status.
- Cancel a BATCH DRAIN or BATCH STOP command that has not been completed yet.
- Have a task machine run a specified job, regardless of any dispatching constraints on the job (such as whether the start window is open or not).
- Remove all dispatching constraints on a job, so that it runs on the next suitable task machine, subject only to the facility priority mechanisms.

```
┌─ Format ──────────────────────────────────────────────────────┐
│                                                                │
│  ►►──BATCH──START─┬──────────────────────────────────┬──►◄     │
│                   ├─ALL──────────────────────────────┤         │
│                   ├─taskid───────────────────────────┤         │
│                   ├─taskid NEXT──────────────────────┤         │
│                   ├─taskid─┬─userid─┬─┬─nodeid─┬─jobid─┤        │
│                   │        └─*──────┘ └─*──────┘       │        │
│                   └─┬─userid─┬─┬─nodeid─┬─jobid────────┘        │
│                     └─*──────┘ └─*──────┘                       │
│                                                                │
└────────────────────────────────────────────────────────────────┘
```
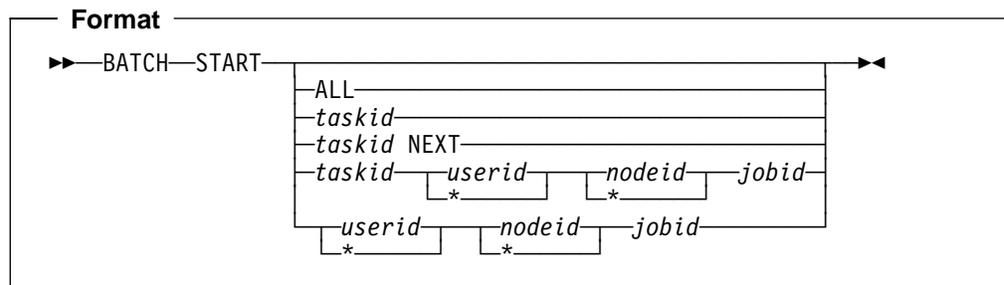
no option

Enter BATCH START without any option to cancel a BATCH STOP command. A BATCH STOP command will be canceled only if the control program in the BATCH monitor machine has not yet stopped.

ALL

Use this option to return all task machines that are in HOLD or OFFLINE status to normal operation. BATCH START ALL also cancels a BATCH DRAIN command that has not yet been completed.

Task machines are put in HOLD status by the BATCH DRAIN and BATCH FORCE HOLD commands. Task machines are put in OFFLINE status if the VM Batch Facility was unable to AUTOLOG the machines to run jobs.

Note: Be sure that you have corrected the problem with an OFFLINE task machine before you restart it. Otherwise, the VM Batch Facility will mark the task machine OFFLINE again when it is unable to AUTOLOG the task machine.

taskid

To return a single task machine from HOLD or OFFLINE status to normal operation, enter the user ID of the task machine. BATCH START *taskid* also cancels a BATCH DRAIN *userid* command that has not been completed.

Note: The task machine user ID is shown as *taskid* in the format illustration to avoid confusion with the job owner user ID option.

taskid NEXT

Use this option to have a task machine run the next job that is eligible to start on it. The task machine can be in READY, HOLD, OFFLINE, or WINDOW NOT OPEN status.

When the job is completed, an OFFLINE task machine is reset to READY, HOLD, or WINDOW NOT OPEN status, whichever is appropriate.

Other task machines will resume the status that they ordinarily would have. HOLD task machines return to HOLD. READY and WINDOW NOT OPEN task

machines resume READY, WINDOW NOT OPEN, or HOLD status, depending on their start time windows and whether a BATCH DRAIN command has been entered.

For *taskid*, enter the user ID of the task machine that is to run the job.

Note: Be sure that you have corrected the problem with an OFFLINE task machine before you use it to run a job. Otherwise, the VM Batch Facility will mark the task machine OFFLINE again when it is unable to AUTOLOG the task machine.

taskid userid nodeid jobid

Use these options to have a task machine that is in HOLD, READY, OFFLINE, or WINDOW NOT OPEN status run the job that you specify.

The job will be started regardless of any of the following normal dispatching constraints:

- The job's dispatching priority in relation to other dispatchable jobs.
- Whether the job's class is compatible with the task machine that you specify.
- Whether the job's window is open or not.
- Whether the task machine's window is open or not.
- Whether the job has been held by a HOLD command.
- Whether the job is waiting for a chain condition to be satisfied.
- Whether the job is waiting for a synchronous submission to complete.
- Whether the job is subject to a REPEAT HOLD condition.
- Whether the job owner has more jobs running than the limit set in the CONTROL FILE.
- Whether the job is subject to LLS and system resource contention levels are preventing LLS subject jobs from being started. However, if the job is LLS subject it may well be suspended after starting.

The job is however subject to possible rejection by the job screening installation exit (DGRUJB EXEC). Recurring jobs do not require confirmation by the job recurrence exit (DGRECUR EXEC).

The job will run under its current PRINT, PUNCH, and SECONDS limits. And the task machine will be autologged with the virtual storage size set for the job when it was received.

When the job is completed, an OFFLINE task machine is reset to READY, HOLD, or WINDOW NOT OPEN status, whichever is appropriate.

Other task machines will resume the status that they ordinarily would have. HOLD task machines return to HOLD. READY and WINDOW NOT OPEN task machines resume READY, WINDOW NOT OPEN, or HOLD status, depending on their start time windows and whether a BATCH DRAIN command has been entered.

For *taskid*, specify the user ID of the task machine that should run the job. You can select any task machine that is in HOLD, READY, WINDOW NOT OPEN, or OFFLINE status.

For *userid*, specify the user ID of the owner of the job that is to be run. You can specify an asterisk (*) to refer to your own userid.

For *nodeid*, specify the node ID from which the job was submitted. You can specify an asterisk (*) to refer to the node you are logged on to.

For *jobid*, specify the jobid of the job that is to be run. To review the user's jobs, enter:

```
batch status userid all
```

Note: Be sure that you have corrected the problem with an OFFLINE task machine before you use it to run a job. Otherwise, the VM Batch Facility will mark the task machine OFFLINE again when it is unable to AUTOLOG the task machine.

userid nodeid jobid
Use these options to have the job that you specify run on the next suitable task machine.

The job will be started regardless of any of the following normal dispatching constraints:

- Whether the job's window is open or not.
- Whether the job has been held by a HOLD command.
- Whether the job is waiting for a chain condition to be satisfied.
- Whether the job is waiting for a synchronous submission to complete.
- Whether the job is subject to a REPEAT HOLD condition.
- Whether the job owner has more jobs running than the limit set in the CONTROL FILE.
- Whether the job is subject to LLS and system resource contention levels are preventing LLS subject jobs from being started. However, if the job is LLS subject it may well be suspended after starting.

The job will not necessarily run on the next available task machine since the following dispatching constraints do still apply:

- The job's dispatching priority in relation to other dispatchable jobs.
- Whether the job's class is compatible with the available task machine.
- Whether the task machine's window is open.

The job is also subject to possible rejection by the job screening installation exit (DGRUJB EXEC), but a recurring job will not require confirmation by the job recurrence exit (DGRECUR EXEC).

The job will run under its current PRINT, PUNCH, and SECONDS limits. The task machine will be autologged with the virtual storage size set for the job when it was received.

Examples

To cancel a BATCH STOP command that has not yet stopped the control program in the monitor machine, enter:

```
batch start
```

To have a task machine named BATCH1 start CHARLIE's job URGENT, enter:

```
batch start batch1 charlie * urgent
```

For the command to work, BATCH1 must not be working on any other job.

Usage Notes

1. **Jobs affected by this command:** This command only affects jobs that have been accepted by the VM Batch Facility monitor machine. Jobs that are

submitted to, but not accepted by the VM Batch Facility are not affected by this command.

2. **Verifying that the command has been completed:** The VM Batch Facility will send a message indicating that your BATCH START command has been executed.

3. **Tasks in OFFLINE status:** When the VM Batch Facility determines that a task machine is defective, it marks the task machine as OFFLINE to prevent the task machine from being selected to run any more jobs. Because the initiation of a task machine to run a job constitutes some processing overhead, this prevents time and resource being wasted on ineffective starts of a malfunctioning task machine.

   Any problem that has caused a task machine to be put offline would be reported on the monitor console. The reported problem should be corrected before attempting to reactivate the task machine.

   Problems that can cause task machines to be put in OFFLINE status are:

   - Mismatch between the logon password recorded for a task machine in the CONTROL FILE, the CP directory entry of the task machine, or in the Resource Access Control Facility (RACF).

   - Storage requirement specified in the job class entry exceeded the maximum virtual storage defined for the task machine. This problem will recur until one of these actions is taken:

      a. The maximum virtual storage of the task machine is increased.

      b. The job class virtual storage requirement is reduced.

         If the job class requirement is lowered, all jobs queued in the class when the change is made should be canceled and resubmitted. Those jobs would be started with the old virtual storage requirement of the job class, and would cause the task machine to be marked as OFFLINE again.

   - The task machine has no access to the VM Batch Facility task and user interface software. If the job has never been able to run any job, then this is a configuration problem. If it happens only with a particular job, it may be that the job is releasing the disk on which the VM Batch Facility software resides.

   When the problem has been corrected, you may re-activate the task machine by any one of the following actions:

   - Issuing the BATCH START command, specifying the task machine that is to be re-activated.

   - Issuing the BATCH CONTROL SET command.

   - Stopping and restarting the monitor.

BATCH STATUS: Reviewing Job Status

Use the BATCH STATUS command to review the status of any VM Batch Facility job.

An administrator may use the CLEAR option of the BATCH STATUS command to clear job information for any user.

```
┌─ Format ──────────────────────────────────────────────────────────────┐
│                         ┌─*──*──ALL──────────────────────────────┐     │
│  ►►──BATCH──STATus──────┼────────────────────────────────────────┼──►◄ │
│                         │  ┌─*──────┐  ┌─*──────┐  ┌─Active─┐     │     │
│                         └──┼─userid─┼──┼─nodeid─┼──┼─ALL────┤           │
│                            └─ALL────┘  └─ALL────┘  ├─Clear──┤           │
│                                                    ├─Hold───┤           │
│                                                    │      ┌─jobid─┐     │
│                                                    │  └─JOB─┘             │
│                                                    ├─Old────┤           │
│                                                    ├─Queued─┤           │
│                                                    ├─REPeats┤           │
│                                                    ├─Running┤           │
│                                                    └─Wait───┘           │
└────────────────────────────────────────────────────────────────────────┘
```

The BATCH STATUS command is described in detail in the *VM Batch Facility User's Guide*.

BATCH STOP: Stopping the VM Batch Facility

Use the BATCH STOP command to shut down the VM Batch Facility. The command stops the VM Batch Facility control program in any of three modes:

- Graceful shutdown. BATCH STOP without any options stops the VM Batch Facility control program only when all currently running jobs have been completed or canceled. In this mode, the monitor machine continues to monitor running jobs and collect accounting data. No new jobs will be started.

- Stop the control program in the monitor machine, but leave jobs running. BATCH STOP NOW stops the control program right away, but lets task machines finish jobs that are currently running. In this case, task machines run without any resource limit checking by the monitor machine.

- Immediate shutdown. BATCH STOP JOBS cancels all running jobs, logs off all task machines, and stops the control program in the monitor machine right away. ERROR option processing is not performed for canceled jobs. Other normal job completion processing is performed.

```
┌─ Format ────────────────────────────────────────────────┐
│                                                          │
│  ►►──BATCH STOP─────────────►◄                           │
│                  ├─NOW──┤                                │
│                  └─JOBS─┘                                │
│                                                          │
└──────────────────────────────────────────────────────────┘
```

no option

Enter BATCH STOP without any options to have the VM Batch Facility shut down gracefully. The command causes the following to occur:

1. The monitor machine and task machines continue running normally until all currently running jobs are finished.
2. Task machines log off as they finish their current jobs. The VM Batch Facility does not start new jobs in task machines.
3. The control program in the monitor machine stops running after the last job is completed or canceled.

NOW

Enter the NOW option to stop the control program in the monitor machine right away, but let task machines continue with any jobs currently running.

Stopping the control program before task machines finish their current jobs has the following effects:

1. Currently running jobs are not subject to resource usage limitations. Task machines will continue running jobs even if resource limits are exceeded.
2. Stalled jobs will not be detected and will not be automatically canceled by the VM Batch Facility.
3. If the job accounting exit is in use, accounting records will be created for jobs that are completed after the control program stops. However, the records will not be accurate because they will reflect the last data available to the control program before it stopped. The accounting records will be generated when the VM Batch Facility is restarted.
4. If any jobs left running after a STOP NOW command issue a JOBSTEP command while the monitor is down, the JOBSTEP command will fail.
5. Jobs that are suspended at the time the control program is is stopped remain suspended.

6. Load level scheduling ceases with the cessation of the monitor. Jobs that have been suspended by the LLS remain suspended.

If the control program is restarted before current jobs have been completed, the VM Batch Facility resumes normal monitoring and accounting of the running jobs.

JOBS

Use this option to shut down the VM Batch Facility immediately, canceling any running jobs. The command will cause the following to occur:

1. All running jobs are canceled and will not be restarted when you restart the VM Batch Facility. The VM Batch Facility lets job owners know that their jobs have been canceled by sending them a message. Accurate accounting records are created for the canceled jobs if the accounting exit is in use.
2. The ERROR job control option, if specified, is ignored when jobs are canceled.
3. The control program in the monitor machine stops and all task machines log off as soon as the command is processed.
4. Recurring jobs are rescheduled.
5. Only CHJOB commands should be issued by installation exits when processing the cancellation of jobs in response to a BATCH STOP JOBS command. All other commands from exits are ignored at this time.

Examples

To shut down the VM Batch Facility gracefully, enter:

```
batch stop
```

The VM Batch Facility will continue normal operation–except that new jobs will not be started–until the last currently running job has ended.

To shut down the VM Batch Facility immediately, canceling any running jobs, enter:

```
batch stop jobs
```

No ERROR job control option processing will be done for the canceled jobs.

Usage Notes

1. **Verifying that the command has been accepted:** You will receive a message acknowledging that a BATCH STOP command has been received.

2. **Canceling a BATCH STOP command:** To cancel a BATCH STOP command if the control program has not yet stopped running, enter:

   ```
   batch start
   ```

3. **Jobs submitted after stop:** The VM Batch Facility continues to accept and queue new jobs after a BATCH STOP command until the control program stops running. No new jobs will be started.

   Jobs submitted after the control program has stopped are held in the monitor machine reader until the VM Batch Facility is restarted.

4. **Queued jobs:** Queued jobs are not affected by any BATCH STOP command. Entering a BATCH STOP command will not cause the VM Batch Facility to lose any jobs that are queued to run.

5. **Suspended jobs:** A suspended job will prevent a BATCH STOP command from being completed. Also, a suspended job remains suspended and inactive after a BATCH STOP NOW command.

6. **CLASS AUTOLOG jobs:** BATCH STOP commands do not affect CLASS AUTOLOG jobs that have already started running on job owners' virtual machines.

7. **Jobs left running:** Jobs left running by a BATCH STOP NOW command, and jobs canceled by a BATCH STOP JOBS command, do not have their task machines cleaned when they end. There may be material left in spool files and on the 191 disks. They are cleaned when the monitor is restarted.

BATCH SUSPEND: Temporarily Halting Jobs

Use this command to temporarily halt a running task machine without causing the job it is running to stall or be canceled.

```
┌─ Format ──────────────────────────────────────────────────┐
│  ►►──BATCH SUSpend──taskid──►◄                             │
└───────────────────────────────────────────────────────────┘
```

taskid

Enter the user ID of the task machine that is to be temporarily halted. The command will cause the task machine to:

- Stop running its current job.
- Stay logged on but inactive until you release it with the BATCH RESUME command or cancel the job with BATCH CANCEL, BATCH FORCE, or BATCH STOP JOBS.

```
┌─ Example ─────────────────────────────────────────────────┐
│                                                            │
│  To suspend user CHARLIE's job STALLJOB, enter these commands: │
│                                                            │
│  1. This command finds out which task machine STALLJOB is running in: │
│                                                            │
│     batch status charlie stalljob                          │
│                                                            │
│     The job status message will include the user ID of the task machine if the │
│     job is still running.                                  │
│                                                            │
│  2. This command suspends the task machine and STALLJOB.  For userid, │
│     specify the user ID of the task machine that is running STALLJOB: │
│                                                            │
│     batch suspend userid                                   │
│                                                            │
│  The task machine will stop executing the job.  It will remain logged on and │
│  suspended until one of these commands is entered:         │
│                                                            │
│  1. The BATCH RESUME command to restart the task machine and the halted │
│     job.                                                   │
│  2. BATCH FORCE or BATCH CANCEL to terminate the job and free the task │
│     machine for another job.                               │
│  3. BATCH STOP JOBS to shut down the VM Batch Facility.    │
│                                                            │
└────────────────────────────────────────────────────────────┘
```

Usage Notes

1. **Verifying that the command has been completed:** The VM Batch Facility will send you a status message for the suspended task machine, showing that the command has been completed.

2. **Job integrity:** The BATCH SUSPEND command does not affect the integrity of a job. When you enter the BATCH RESUME command, the task machine starts running the job at the point at which the job was suspended.

3. **VMCF Usage:** A job that uses VMCF, steals the external new PSW, or that modifies the CMS environment in any way, may inhibit the SUSPEND and RESUME function for the duration of that job.

4. **LLS Usage:** Jobs suspended by entering this command will not be resumed by the Load Level Scheduler.

Chapter 5. VM Batch Facility Installation Exits

The VM Batch Facility provides exits that you can change to tailor the facility's operation.

Each exit is provided with information relevant to its purpose, and some exits are provided with the means to affect execution of the facility.

Sample exits are supplied with the product. The facility will run normally if the supplied exits are not modified; however it is likely that you will want to change them.

The prologs of the sample exits contain detail instruction on the usage of each exit. Each sample exit also contains code illustrating how to access the information provided to the exit, and examples of possible exit function.

The exit prologs are reproduced at the end of this chapter, starting on page 135.

Exits are provided to the following VM Batch Facility components:

1. The VM Batch Facility monitor invokes the following exits which run in the monitor virtual machine:

   Job accounting exit (DGRUAC EXEC)
   DGRUAC EXEC provides an opportunity to perform job accounting.

   It is called when a running job ends for any reason, and facility accounting has been enabled by setting the ACCOUNT keyword in the CONTROL FILE to 1 or 3.

   It is provided with information relevant to the amount of system resource that was used by the job.

   As supplied with the product, it generates an accounting record for each job that is completed. The accounting records are stored in the DGRUAC RECORDS file on the monitor machine 195 minidisk.

   It can be modified to generate accounting records in any form using part or all of the job information that the VM Batch Facility provides to the exit.

   Command screening exit (DGRUCD EXEC)
   DGRUCD EXEC provides an opportunity to screen all incoming commands, including job submissions.

   It is called whenever the monitor machine receives an external command; before the command is executed.

   As supplied with the product, it permits the VM Batch Facility to execute all commands according to standard authorization rules; but rejects all commands from remote nodes, Remote Job Entry (RJE) workstations, and VM Batch Facility task machines.

   It can be modified to provide command screening services such as the following:

   - Let users at remote nodes or RJE workstations enter commands.
   - Allow spawned commands from task machines.
   - Reject VM Batch Facility commands to limit the command authority of job owners or authorized users.

- Let users who are not authorized users enter authorized user commands.

Job screening exit (DGRUJB EXEC)

DGRUJB EXEC provides an opportunity to screen the job control options of all incoming jobs, and to influence their execution.

It is called at various points in a job's normal processing cycle, such as when a job is received, changed, started and completed.

It is complemented by the job problem exit, DGRIDO EXEC, which is called when an abnormal event takes place, such as a cancellation.

As supplied with the product, the exit allows all jobs except those from remote nodes and RJE workstations to be processed normally. Jobs from remote nodes and RJE workstations will be rejected until you modify the exit. It also issues messages notifying the job owner of its passage through the stages of receipt, start and completion.

It can be modified to provide services such as the following:

- Permit a job from a remote node or RJE workstation.
- Provide an alternate user ID for a job from a remote node or RJE workstation.
- Reject a job.
- Confirm that job control options required by your installation were supplied for a job.
- Alter a job's job control options.
- Reject a BATCH CHJOB command without affecting the original job control options.
- Examine any LOCJCL cards that were included with the job.

Job problem handling exit (DGRIDO EXEC)

DGRIDO EXEC provides an opportunity to monitor abnormal events in a job's execution, and to influence the monitor's response to the event.

It is called when an event occurs that has had, or may yet have, a detrimental effect on the processing of a job; such as when a job has failed to start, or has stalled, or has been canceled, among others.

As supplied with the product, it provides no special function.

It can be modified to provide services such as:

- Cancel a job.
- Permit a job that has exceeded its resource usage limits to continue processing.
- Permit a job that has stalled to continue processing.
- Requeue a job that has not started.
- Directly set an alternate user ID.
- Alter a job's job control options.

Message screening exit (DGRUMSG EXEC)

DGRUMSG EXEC provides an opportunity to screen messages issued by the monitor to users.

It is called before any message generated by the monitor is sent to a user.

It is not invoked for messages generated by any of the exits, nor for messages that do not address a user. It is not invoked for messages

generated by the user interface modules, which do not run in the monitor virtual machine.

As supplied with the product, the exit allows the VM Batch Facility to send all messages it issues.

It can be modified to provide message screening services such as the following:

- Replace a message to be issued by the VM Batch Facility with a message specific to your installation.
- Redirect a VM Batch Facility message to a specific user.
- Suppress the sending of a message.

Job recurrence exit (DGRECUR EXEC)
DGRECUR EXEC provides opportunities to manage the scheduling of recurring jobs.

It is called whenever an event occurs that may affect the scheduling of a recurring job, such as when an instance of a job completes and its next run is to be scheduled.

The job recurrence exit will provide the site defined recurrence patterns when queried by the user.

As supplied with the product, it provides no special function.

It can be modified to allow jobs to recur on a regular basis and according to any recurrence pattern that you wish to define.

2. The VM Batch Facility user interface invokes the following exit which runs in the user virtual machine:

User interface exit (DGREVR EXEC)
DGREVR EXEC provides an opportunity to monitor VM Batch Facility commands as they are issued in the user virtual machine.

It is called whenever a VM Batch Facility command is issued by a user.

As supplied with the product, it provides no special function.

It can be modified to provide services such as the following:

- Set preferred default options to the BATCH SUBMIT command.
- Include installation defined job control cards, called LOCJCL cards, with any submitted jobs.

  The LOCJCL cards are made available at job receipt time to the receiving monitor's job screening exit (DGRUJB EXEC), and are also passed to the Facility task control exit (DGROOLY EXEC).

3. The VM Batch Facility task interface invokes the following exit which runs in the task machine that executes a job.

Task control exit (DGROOLY EXEC)
DGROOLY EXEC provides an opportunity to perform any site required processing on the virtual machine in which a VM Batch Facility job is to run.

It is called whenever a task machine is initialized in preparation for the execution of a job. This may be for the initial start of a job, or a restart, or for the commencement of error processing.

At initial start, it is supplied with any LOCJCL cards that were included with the job.

As supplied with the product, it provides no special function.

It can be modified to provide services such as CP accounting according to the information on the LOCJCL cards.

Changing the Exits

The following are the requirements, suggested techniques and restrictions involved in changing exits.

Refer to the *Program Directory* for instructions on how to update the exits.

Monitor exits must be written in REXX.
The monitor exits must be written in REXX so that they can accept extended parameter lists (extended PLISTS). Some of the arguments passed to the exits exceed 8 bytes.

Exit names cannot be changed.
If you replace an exit, you must keep the original name:

DGRUAC EXEC   Job accounting exit
DGRUCD EXEC   Command screening exit
DGRUJB EXEC   Job screening exit
DGRIDO EXEC   Job problem exit
DGRUMSG EXEC   Message screening exit
DGRECUR EXEC   Job recurrence exit
DGREVR EXEC   User interface exit
DGROOLY EXEC   Task control exit

All monitor exits must be available to the monitor.
If any monitor exit is missing at monitor startup, the monitor terminates. If any monitor exit is missing when refreshing exits, then no exits are refreshed.

The User interface and Task control exits are optional.

Monitor exits must be stored on the monitor machine 199 minidisk.
The monitor will only load exits from its 199 minidisk. Monitor exits cannot be stored on any other minidisk.

User and task interface exits must be stored on a common disk.
In order to make these exits available to all users, they must be placed on a disk that is accessible to all users, and they should have a *filemode* number of 2. Task machines become users too if job spawning is permitted on your system, so make sure that the user interface exit is accessible to your task machine as well.

Monitor exits may be changed while a monitor is running.
Monitor exits may be refreshed without taking a monitor down, by issuing the BATCH RELOAD command . Refer to "BATCH RELOAD: Refreshing Installation Exits" on page 110.

Monitor exits must allow for a call with a single parameter of '?'.
Each monitor exit is called with a single parameter of '?' at monitor startup and whenever the monitor exits are refreshed by a BATCH RELOAD command. If you wish to run different versions of a monitor exit, you should take advantage of the call to display a message on the monitor console identifying the version of the exit.

Exits cannot issue standard VM Batch Facility commands.
You cannot enter standard VM Batch Facility commands from within an exit.
However, the DGRUJB,DGRIDO and DGRECUR exits are provided with a
capability to issue some VM Batch Facility commands to the monitor; but only
using a particular method. See "Issuing Commands from VM Batch Facility
Exits" on page 126 for a description of this method.

Exits must issue messages on their own.
Exits cannot use VM Batch Facility message issuing facilities. If an exit is to
issue messages, it must include code to send the messages. The supplied
DGRUJB exit has sample code for issuing messages.

Exits must use only designated return codes.
The VM Batch Facility will accept only specified return codes from each exit.
Other return codes are treated as the default and may generate alert messages.
(Exception: if the job accounting user exit, DGRUAC EXEC, issues a nonzero
return code, the VM Batch Facility will shut down.)

Exits cannot call other programs that load into the user area.
An exit cannot call another program, command, REXX function and so on,
which loads in the user area (X'20000').

**Monitor exits cannot call other programs or execs that have names beginning
with 'DGR'.**

Exits should ensure the integrity of data files.
An exit that writes data to a file is responsible for the integrity of the data in the
case of a system abend. The monitor only maintains its own open files.

Exits should take care when using the program stack.
The Facility often uses the program stack as a communication area with the
exits. The use and sensitivity of the program stack is described in the prolog for
each exit.

The monitor clears the stack after each monitor exit call.

Job submittor type field of the installation exit parameter JOBFLAGS
The JOBFLAGS parameter as supplied to the DGRUJB, DGRIDO and
DGRUAC installation exits, indicates the type of user who submitted the job.
This distinction is made primarily so that the VM Batch Facility and the
installation exits know the correct method of communicating with the job
submittor, for example, by spool files or CP MSG.

There are three types of job submittor: RJE, local and network. The type of job
submittor is determined as follows:

RJE   An RJE user is one who creates a job using the direct interface,
specifying *RJE on the SUBMIT card, indicating to VM Batch Facility
that the job is to be treated as originating from a real RJE device.
Messages should be issued via spool files.

Local   A local user is one who submits a job to a VM Batch Facility monitor
on the same node as the user. Messages should be issued to a
local user by the CP MSG or CP MSGNOH command.

Network A network user is one who submits a job to a VM Batch Facility
monitor on a different node from the user. Messages should be
issued to a network user via RSCS.

Issuing Commands from VM Batch Facility Exits

The job screening exit (DGRUJB EXEC), the job problem exit (DGRIDO EXEC) and the job recurrence exit (DGRECUR EXEC) may affect job processing by the monitor by placing selected VM Batch Facility commands on the program stack. This capability is in addition to the influence the exits may exert through return codes.

They may not issue VM Batch Facility commands through the normal user interface.

No other VM Batch Facility exits may issue VM Batch Facility commands.

These commands are retrieved from the stack by the monitor on return from the exit, and are executed when the monitor has completed its current activity, before any new external commands are processed, and after the return code has been acted on.

Restrictions on Commands Issued by VM Batch Facility Exits

Exit commands are subject to the following restrictions:

- Commands may only be issued by the DGRUJB, DGRIDO and DGRECUR exits. They are ignored when issued by other monitor exits. They will cause facility malfunction if issued from the user interface (DGREVR) or task control (DGROOLY) exits.

- Commands are in the same format as normal VM Batch Facility commands except that they are **not** prefixed with BATCH.

- All commands are in their authorized form. That is, the target job must be explicitly identified by its owning user ID, node ID and job name. No placeholders may be used.

  All commands are assumed to have administrator authority.

- Commands must be coded in uppercase.

- Only the following commands may be issued from the exits:

  - ALTID

  - CANCEL

  - CHJOB

    Note: If the ALTID option is used, then successful execution of the CHJOB command confirms the alternate user ID as well as setting it. This occurs only when a CHJOB command is issued by an exit.

  - DRAIN

  - FORCE

  - FREE

  - HOLD

  - LLS

  - PRIORITY

  - START

> **Note:** The START command may not be issued to cancel a STOP command.

- No more than five commands may be issued by an exit at any one call. If more than five are issued, then all commands issued for the call are discarded.

- Exits must ensure that on their return the stack contains only the required commands.

- All commands from exits except CHJOB are discarded after the command BATCH STOP JOBS has been issued (to terminate the monitor immediately).

- No indication of the success or failure of a command is provided to an exit. Commands and their resulting actions are logged to the monitor console.

  When error messages are generated in response to a command from an exit, the Facility alertee is notified.

- Commands issued by exits are ignored when issued in response to a '?' exit call.

- Only CHJOB commands are accepted when processing the cancellation of jobs after a BATCH STOP JOBS command has been issued.

Although the DGRUJB, DGRIDO and DGRECUR exits are called for a particular job, they may issue commands against any other job if required.

Commands issued by an exit will not invoke any exits during their processing.

For example, the following REXX statement

```
queue 'ALTID' jobuser jobnode jobname jobaltuser
```

places a command on the stack for the monitor to set and confirm the alternate user ID *jobaltuser* for the job defined by *jobuser jobnode jobname*.

Enabling a Facility for Remote Processing

If you have an RSCS link to a remote processing node or RJE workstation, you must tailor your exits to allow jobs and commands from remote users to be processed.

The following items are mandatory:

- The sample command screening exit, DGRUCD EXEC, as shipped will reject remote commands. An example of how to alter the exit is provided in the sample.

- The sample job screening exit, DGRUJB EXEC, as shipped will reject remote jobs. An example of how to alter the exit is provided in the sample.

- When a job is received from the local node, the monitor by default sets the alternate user ID for the job as the user ID of the job submittor.

  For jobs from other nodes, no such default is available. The Facility administrator is required to enter an ALTID command to provide an alternate user ID, before the job is started. This requires that the job be prevented from starting to give the administrator time to enter the command. Also, local users requesting an alternate user ID other than their own must have these requests confirmed by an administrator.

Two automatic mechanisms are provided by the Facility for setting and confirming alternate user IDs; requiring no manual administrator intervention.

1. When a job is received, the DGRUJB exit is called, and it is provided with the alternate user ID requested by the job submittor, and a flag indicating whether confirmation is required. If you wish the request to be honored, or to explicitly set the alternate user ID, then have the exit issue the ALTID command to the monitor at this time.

   This mechanism is recommended.

   Remember, that if a job's suggested alternate user ID is changed by a CHJOB command with the ALTID option, then the exit should re-issue the ALTID command at CHJOB time to confirm the alternate user ID.

2. When the monitor selects a job for dispatch, and the alternate user ID has not been confirmed by an ALTID command (if required), then the monitor does not set an alternate user ID for the job, and takes no further responsibility for the alternate user ID. Rather, it calls exit DGRIDO EXEC with reason = RJEALTU, and this exit now has the opportunity to explicitly set the alternate user ID itself; by issuing the CP DIAGNOSE X'D4'. At the time of this call, the task machine that runs the job has already been autologged.

   Note that the CP DIAGNOSE X'D4' cannot be issued directly by a REXX EXEC. It must be issued from within a module that you create, which may in turn be invoked by DGRIDO EXEC. Remember too, that any module invoked by an exit must not load into the CMS user area, otherwise it will overlay the monitor itself.

   Jobs that have an alternate userid set by this method will not be able to spawn jobs, or issue any VM Batch Facility commands at all.

Job submittors request alternate user IDs by way of the ALTID option of the SUBMIT command. If the ALTID option is not specified, it defaults to their own user ID.

If no alternate user ID is set for a job, the job runs only with the authority of the task machine itself.

When an alternate user ID is confirmed for a recurring job, it remains confirmed for all subsequent occurrences of the job.

Enabling a Facility to Accept Spawned Jobs and Commands

VM Batch Facility commands that are issued from within an VM Batch Facility job are termed "spawned commands", and resultant jobs are referred to as "spawned jobs".

Usage of spawned commands is described in the User's Guide.

The monitor exits DGRUCD, DGRUAC, DGRUJB, DGRIDO and DGRUMSG are provided with a flag indicating whether the subject of an exit call is related to a spawned command or job.

If you wish to permit spawned jobs and commands to a monitor you must tailor the command screening exit (DGRUCD EXEC). The sample exit provided with the product contains active code to prevent acceptance of spawned commands.

You may also consider tailoring the message screening exit (DGRUMSG EXEC) to suppress some messages pertinent to spawned jobs and commands.

Be aware that when VM Batch Facility commands are to be issued from within a job EXEC, the VM Batch Facility user interface must be available to the task machine, together with the user interface exit (DGREVR EXEC) and the VMBATCH DEFAULTS file, if these are in use at your installation.

Note: Jobs that run under an alternate userid that is not known to the VM Batch Facility monitor controlling the job will not be able to spawn jobs and commands; for example, when a job's alternate userid is set during a call to the job screening exit (DGRUJB EXEC) with reason = RJEALTU.

Holding and Freeing Jobs by Exits

Jobs may be held for later execution and subsequently freed by a command from the DGRUJB and DGRIDO exits; or by return code from the DGRUJB exit.

A job can only be held before it actually starts. Generally, this means that if a job is to be held by an exit it must be done by DGRUJB when called for reasons RECVD, CHJOB, or START.

A job so held must be freed before it can run. When the monitor dispatcher selects a job held by an exit, the DGRUJB exit is called for reason HELD to provide the exit with an opportunity to free the job.

Jobs may also be freed or held by administrator command. Administrator generated HOLD and FREE commands take precedence over exit commands:

- A job freed by an administrator cannot be held by an exit.

- A job held by an administrator cannot be freed by an exit.

- Jobs held or freed by an exit can be freed or held by an administrator.

Local JCL cards (LOCJCL)

You may wish to tailor the facility to perform job control functions that are unique to your installation. The VM Batch Facility provides an ability to include installation defined information in the job control for each job.

- Prepare the user interface exit (DGREVR EXEC) to create local JCL cards (or LOCJCL cards) according to your specifications, and these will be inserted into the job control deck for each job.

- Prepare the monitor job screening exit (DGRUJB EXEC) to provide processing of the LOCJCL cards on the monitor virtual machine when the job is received.

- Prepare the task control exit (DGROOLY EXEC) to provide processing of the LOCJCL cards on the task virtual machine before the job is started in that machine.

The sample DGREVR, DGRUJB and DGROOLY EXECs provided with the product contain examples of how this is done.

Some typical uses:

- An installation uses an accounting code that is longer than the standard eight bytes provided by VM Batch Facility. Such an installation would utilize DGREVR EXEC to include the longer accounting code in a LOCJCL card. When the job is sent to the task machine, DGROOLY EXEC is passed the LOCJCL card and is then able to perform job accounting according to the installation's requirements.

- An installation requires that remote users provide a password admitting them to the use of an alternate userid. The password would be placed on a LOCJCL card by DGREVR EXEC, from where it is retrieved by the monitor's DGRUJB EXEC for verification and confirmation of the use of the requested ALTID.

Creating LOCJCL cards

DGREVR EXEC is called by the VM Batch Facility user interface component, DGRBATCH EXEC, whenever a VM Batch Facility command (other than JOBSTEP or ADDRESS) is entered. Like DGRBATCH EXEC, DGREVR EXEC executes in the invoking user's virtual machine.

When the command is SUBMIT, then DGREVR may create LOCJCL cards as follows:

- The format of a LOCJCL card is

      LOCJCL text

  The first six characters identify the card to be a LOCJCL card, followed by a blank, followed by the text. The length of the text may not exceed 80 characters and the card may not contain any non-displayable characters (for example, X'00', X'15').

  No translation is done on the text, but the LOCJCL keyword is translated to upper case.

- Place the card in the program stack, from where it is retrieved by DGRBATCH and included in the JCL deck for the job.

- Multiple LOCJCL cards may be inserted into a job deck, to a maximum of 3.

  When multiple LOCJCL cards are used, it is recommended that the text of LOCJCL cards contain an identifier to distinguish one card from another.

  Each LOCJCL card must be stacked separately.

This is the only mechanism whereby LOCJCL cards may be included in a job deck. There is no LOCJCL option to any VM Batch Facility command. However, when using the Direct or RJE interface (which does not execute DGRBATCH EXEC), LOCJCL cards may be included among the job control option cards submitted in the job file.

Note that the VM Batch Facility does not require that the DGREVR and DGROOLY exits be available. Their absence does not constitute a notifiable exception. It is your responsibility as an administrator to ensure that these are accessible by your users and task machines as you require. If you have enabled job spawning on a monitor, then DGREVR should be made available to the task machines as well.

In particular, be aware that the DGREVR EXEC cannot be used to provide any tamper proof function, such as might be required for security processing. This is because it runs within a submitting user's virtual machine, and may be easily

replaced or altered by a user. However, LOCJCL cards may be subjected to tamper proof verification by DGRUJB EXEC; DGROOLY EXEC is also tamper proof, but it may reside on a common disk, so it is not private.

No mechanism is provided for altering LOCJCL cards once a job has been submitted; for example, they may not be altered by a CHJOB command.

Defining default job control options for your installation

You may wish to tailor the facility to select default options to the BATCH SUBMIT command other than those provided by the VM Batch Facility. For example, the default for the ERROR option on the BATCH SUBMIT command is 'DUMP', and you may prefer the default option for your users to be 'NODUMP'.

This ability is provided by the VM Batch Facility user interface exit (DGREVR EXEC).

DGREVR EXEC is called by the VM Batch Facility user interface component, DGRBATCH EXEC, whenever a VM Batch Facility command (other than JOBSTEP and ADDRESS) is entered. Like DGRBATCH EXEC, DGREVR EXEC executes in the invoking user's virtual machine.

When the command is SUBMIT, then DGREVR may supply default options to be applied to the command by placing these in the program stack.

- Only SUBMIT command options may be supplied by DGREVR. No other part of the BATCH SUBMIT command, such as a job's *filename*, may be supplied.

- All options are coded using the same syntax as if they had been entered on the command line. For example, the CLASS option would be specified on the command line as

        BATCH SUBMIT TEST (ID TEST CLASS B

  To have 'CLASS B' specified as a default option for a job, DGREVR EXEC would stack

        CLASS B

  before returning control to DGRBATCH.

- Options provided by DGREVR EXEC are default options. They are only applied when an equivalent option is not entered on the command line. They do not override an explicitly requested option.

- It is not necessary for DGREVR EXEC to provide values for all BATCH SUBMIT options. Product provided defaults are appropriately documented.

- Multiple options may be supplied by DGREVR, on a single or on multiple stacked lines.

- Options provided by DGREVR EXEC are subject to the same scrutiny from DGRBATCH EXEC as options provided on the command line.

Be aware that the DGREVR EXEC cannot be used to provide any tamper proof function, such as might be required for security processing. This is because it runs within a submitting user's virtual machine, and may be easily replaced or altered by a user. You may use the monitor job screening exit (DGRUJB EXEC) to verify that any options that are required at your installation have been specified.

See the examples in the sample DGREVR EXEC provided with the product.

Enabling a Facility for Job Recurrence

You may wish to allow jobs to be rescheduled on a recurring basis without requiring users to resubmit them for each run. This ability is provided by the job recurrence exit of the VM Batch Facility, DGRECUR EXEC.

You code the DGRECUR EXEC to provide scheduling support for 'recurrence patterns'. For example, if you wish to provide users with the ability to submit jobs that run every day, you would then code the DGRECUR EXEC to support a recurrence pattern called, perhaps, 'DAILY'. A user may then specify 'DAILY' for the REPEAT option of the BATCH SUBMIT command, to request that when the job has completed, it be rescheduled to run again the following day.

Essentially, the exit provides scheduling support of a recurrence pattern by supplying the date and time for a job's next run. The exit is called to do this each time an instance of the job completes. For a recurrence pattern of 'DAILY', you would code the DGRECUR EXEC to add one day to the current date, and then issue a VM Batch Facility CHJOB command using the resulting date for the DATE option, effectively scheduling the next run of the job for the following day.

The sample DGRECUR EXEC that is provided with the VM Batch Facility contains inactive code illustrating scheduling support for a pattern called 'DAILY'. Use it as a guide in creating support for patterns that you require.

All recurring jobs must be stored on the monitor's 193 Disk, and must not be queued in its virtual punch. You should ensure that enough disk space is provided for recurring job storage. You may consider setting the MAXIMUM-JOB-SIZE keyword in the monitor's CONTROL FILE to zero, effectively reserving the monitor's 193 disk exclusively for recurring jobs. Recurring jobs will be rejected if they are not able to be stored on the monitor's 193 disk.

The job accounting exit (DGRUAC EXEC), the job screening exit (DGRUJB EXEC) and the job problem exit (DGRIDO EXEC) are passed information describing a jobs recurrence. You should alter existing exits to exploit this information as needed.

Types of Recurrence

A job is made to recur *explicitly* by specifying its recurrence pattern using the REPEAT option of the BATCH SUBMIT command. However, a job may also be made to recur *implicitly* by chaining it (directly or indirectly) to an explicitly recurring job. Only explicitly recurring jobs rely on the DGRECUR exit to schedule their next run; implicitly recurring jobs do not have a direct association with a recurrence pattern. At completion they are requeued with their original chain dependency, and will recur again when their chain condition is once more satisfied.

Recurrence Pattern Names

A recurrence pattern is identified by a unique name, consisting of an alphanumeric string of 1 to 8 characters. The following pattern names are reserved:

1. The pattern name 'NEVER' signifies no explicit recurrence.

   A job that is submitted with REPEAT NEVER or without a specification of the REPEAT option will have no explicit recurrence.

2. The pattern name 'HOLD' signifies suspension of recurrence.

3. The pattern name 'NOHOLD' signifies resumption of recurrence.

4. The pattern name 'KEEP' signifies manually scheduled recurrence.

   Such jobs are not discarded on completion. The monitor requeues them with the DATE option set to 'HOLD'. The DGRECUR exit is not invoked to schedule a next occurrence. A next occurrence requires manual scheduling by issuing a BATCH CHJOB command.

   This pattern may be used without any provision of scheduling support in the DGRECUR exit.

Function of the Job Recurrence Exit (DGRECUR EXEC)

The DGRECUR exit is called to manage the automatic recurrence of explicitly recurring jobs. It is not called in relation to implicitly recurring jobs, nor in relation to jobs that do not recur automatically, that is, have the reserved recurrence pattern 'KEEP'.

The parameters that are passed to the exit, and the meaning of all return codes from the exit, are described in the prologue of the sample DGRECUR EXEC provided with the VM Batch Facility. See page 151.

The DGRECUR exit is called by the Facility monitor at the following times:

1. When a SUBMIT or CHJOB command is received specifying a non reserved recurrence pattern name for the REPEAT option, the exit is called with action = 'ACCEPT' to verify the pattern name.

   The exit must reply by setting a return code indicating whether or not the requested recurrence pattern is acceptable. If the pattern is not acceptable, then the command is rejected by the monitor.

2. When an instance of an explicitly recurring job completes (normally or abnormally), the monitor requeues the job with its existing job control options, except that the DATE option is set to 'HOLD'. The DGRECUR exit is called with action = 'NEXT' to schedule the next occurrence of the job.

   The exit must reply to the monitor by placing a CHJOB command on the program stack, specifying the DATE option to schedule the next occurrence of the job.

   If the exit fails to reply to the call, the job remains in DATE HOLD and a message is sent to the owner and the alertee. The next occurrence must then be scheduled manually.

   Note that the job is requeued with its original start time window as set by the WBEGIN and WEND job control options. In general, one would assume that if a user wanted a job to be run at a particular time of day, then the user would set the start time window, perhaps on the original submission. In general it should not be necessary for the DGRECUR exit to specify the WBEGIN and WEND options on the CHJOB commands that it stacks, since the original window definition is passed on to each successive instance.

3. When a job is selected for dispatch by the monitor, but the job is prevented from running by a suspension of recurrence (it has had a 'CHJOB (REPEAT HOLD' issued against it), then the exit is called with action = 'SKIP' to schedule the next occurrence of the job in the same manner as the 'NEXT' call described

above. If recurrence of the job remains suspended then the next and following occurrences will also be skipped.

A job will only be 'skipped' if suspension of recurrence is the *only* impediment to its running. If a job is subject to other hold conditions, for example a BATCH HOLD command, then it is not even selected for dispatch.

4. When a job is selected for dispatch by the monitor, and before the job is started, the exit is called with action = 'CONFIRM' to verify that the actual start of the job remains consistent with the pattern definition; since the actual time that a job gets dispatched may differ from the scheduled start time and date (for example after system outages).

The exit indicates the acceptability of the start by means of a return code.

If the exit provides a return code preventing the job start, then the monitor requeues the job with the DATE option set to HOLD, and the exit must place a CHJOB command on the program stack, specifying the DATE option to schedule the next occurrence of the job. If the exit fails to provide the command, then the job remains queued with DATE HOLD and a message is sent to the owner and the alertee. The next occurrence must be scheduled manually.

5. On receipt of a BATCH QUERY REPEATS command the exit is called with action = 'QUERY'. The exit must reply directly to the user indicating which repeat patterns are available.

6. At monitor start time, and whenever the installation exits are refreshed by a BATCH RELOAD command, the exit is called with action = '?' to identify itself on the monitor console.

Note that if a job instance is forcibly initiated by a BATCH START command, then the DGRECUR exit is not called to CONFIRM a job start; however it is called at the completion of the instance with action = 'NEXT' to schedule the next occurrence.

Occurrence dates scheduled by the DGRECUR exit may be manually overridden or supplemented whenever the job is not running, by issuing a BATCH CHJOB command directed at the job and specifying the DATE option.

Relation to other Monitor Exits

In relation to the existing installation exits:

- When a job or CHJOB is received, the recurrence exit is called for ACCEPTance after the job command screening exit (DGRUCD EXEC), but before the job screening exit (DGRUJB EXEC).

- When a job instance is started, the recurrence exit is called for CONFIRMation before the job screening exit (DGRUJB EXEC).

- On completion of a job instance, DGRUJB EXEC (or the problem exit DGRIDO EXEC) and the accounting exit (DGRUAC EXEC) are called before DGRECUR EXEC.

- No other exits are called when a recurring job is skipped.

Job Accounting Exit (DGRUAC EXEC)

The prolog to the sample job accounting exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                          */
/*  Exec name: DGRUAC                                       */
/*                                                          */
/*          Job Accounting Exit for the VM Batch Facility   */
/*                                                          */
/*  Function:                                               */
/*      1. This exit is called when a running job ends for any  */
/*         reason. It provides the opportunity to perform site  */
/*         dependent job accounting.                        */
/*      2. It is called only when the "ACCOUNT" option in the   */
/*         Control File is set to 1 or 3.                   */
/*      3. It is not called for Class AUTOLOG jobs.        */
/*      4. It will be called at the following job stages. The   */
/*         parameter 'Reason' indicates the stage.         */
/*         a. Reason = JOB                                  */
/*            This happens when the job ends, either normally or */
/*            after being canceled. This call records      */
/*            the jobs resource usage during its run.      */
/*         b. Reason = ERROR                                */
/*            This happens when a job was cancelled, and additional */
/*            processing was required according to the job's */
/*            ERROR option. This call records              */
/*            the jobs resource usage during ERROR processing. */
/*         c. Reason = RESTART                              */
/*            This happens when a job was interrupted by a system */
/*            outage, and is about to restart. This call records */
/*            the jobs resource usage prior to being interrupted. */
/*                                                          */
```

Figure 3 (Part 1 of 3). VM Batch Facility Job Accounting Exit Prolog

```
/*  Entry conditions:                                                */
/*                                                                   */
/*     The following information is provided to DGRUAC (all numeric  */
/*        information is provided in decimal form):                  */
/*                                                                   */
/*     Parameter   Size (bytes)   Description                        */
/*     _____ */
/*     Reason        7            Reason being called: JOB, ERROR or ? */
/*                                - '?' for exit identification      */
/*                                - 'JOB' on job completion          */
/*                                - 'ERROR' at end of ERROR processing */
/*                                - 'RESTART' after a system interrupt */
/*                                                                   */
/*     No other parameters are provided when Reason is '?'           */
/*                                                                   */
/*     Batalert    8            ID of person receiving alert msgs.   */
/*     Jobnode     8            ID of job owner's RSCS Node.         */
/*     Jobowner    8            ID of job owner.                     */
/*     Jobname     8            ID of job.                           */
/*     Jobclass    8            Job class.                           */
/*     Jobacnt     8            Account number for the job.          */
/*     Jobsdate    10           Date job was scheduled to start.     */
/*     Jobupj      1            Job Priority - Set by the user.      */
/*     Jobwb       4            Job start window - open.             */
/*     Jobwe       4            Job start window - close.            */
/*     Jobaltuser  8            Alternate userid as set by monitor   */
/*                              It has the value '*' if no alternate */
/*                              userid was set by the monitor.       */
/*     Jobrecp     8            Recurrence pattern. Set to '*' if    */
/*                                 job does not recur explicitly.    */
/*     Reserved    1            Reserved for IBM use                 */
/*     Jobflags    4            Miscellaneous job information        */
/*                 1st character: User type                          */
/*                                'L' = Local user                   */
/*                                'N' = Network user (remote node)   */
/*                                'R' = RJE device                   */
/*                 2nd character: LLS subjection                     */
/*                                'L' = LLS subject                  */
/*                                'I' = Independent of LLS           */
/*                 3rd character: Job origin indicator               */
/*                                'U' = User originated job          */
/*                                'T' = Task originated job          */
/*                 4th character: Job recurrence indicator           */
/*                                'R' = Recurs explicitly by jobrecp */
/*                                'C' = Recurs implicitly by chain   */
/*                                'N' = Does not recur               */
/*     Jobupe      1            Job Priority - Expedited. (IF ANY)   */
/*     Jobstep     8            Last recorded job step.              */
/*     Jobmch      8            Task machine id job ran on.          */
/*     Jobqtime    18           Date/Time job submitted.             */
/*                              (Date/Time re-queued if recurring.)  */
/*     Jobrtime    18           Date/Time job (or error option) began. */
/*     Jobntime    18           Date/Time job (or error option) ended. */
/*                                                                   */
```

Figure 3 (Part 2 of 3). VM Batch Facility Job Accounting Exit Prolog

```
/*  The following resource use counters are reset before each start   */
/*  (or restart) of a job, and before ERROR processing. Therefore      */
/*  each accounting record, whether JOB, ERROR or RESTART, makes a     */
/*  contribution to the total resource used  by the job.               */
/*                                                                     */
/*      Jobconn      10          Task machine connect time (hhhh:mm:ss)*/
/*      Jobttime     6           Total CPU Time (in seconds).          */
/*      Jobvtime     6           Total Virtual CPU Time (in seconds).  */
/*      Jobrdr       6           Total Reader records.                 */
/*      Jobprint     6           Total Print records.                  */
/*      Jobpunch     6           Total Punch records.                  */
/*      Jobpgr       6           Total Page reads.                     */
/*      Jobpgw       6           Total Page writes.                    */
/*      Jobsio       6           Total Non-spooled SIO's.              */
/*      Jobwss       6           Maximum observed adjusted WSS.        */
/*      Jobsuspn     2           Number of times job suspended.        */
/*      Jobstlct     2           Number of times job stalled.          */
/*      Jobvvect     6           Total user vector time (in seconds).  */
/*      Jobtvect     6           Total user + CP vector time (in secs) */
/*                                                                     */
/*  Exit conditions:                                                   */
/*      A nonzero return code will shut down the VM Batch Facility.    */
/*                                                                     */
```

Figure 3 (Part 3 of 3). VM Batch Facility Job Accounting Exit Prolog

Sample Accounting Record

The following diagram shows the format of a single sample accounting record produced by the DGRUAC EXEC, the job accounting exit. The sample record assumes that the exit has not been modified and is at it was when shipped to you.

Job accounting records are stored in the DGRUAC RECORDS file on the monitor machine 195 minidisk.

```
┌─ Sample Job Accounting Record ─────────────────────────────────────────

Sample record 1:  job cost record

JOB PENN 095758 A * ALLDONE VMSYS1 BATCH1 01/09/95 01/09/9509:57:58
01/09/9509:58:01 01/09/9509:58:08 9 9 0000 2400 0000:00:00 000017 000017
000000 000000 000018 000034 000000 000051 000028 01 00
PENN NIUR 000000 000000 DAILY

Sample record 2:  ERROR job control option processing record

ERROR PENN 095758 A * * VMSYS1 BATCH1 01/09/95 01/09/9509:57:58
01/09/9509:58:11 01/09/9509:58:18 9 9 0000 2400 0000:00:00 000000 000000
000018 000000 000000 000033 000000 000046 000035 00 00
PENN NIUR 000000 000000 DAILY
```

The accounting records in DGRUAC RECORDS contain all parameters passed to DGRUAC EXEC except *Batalert*, and the reserved field.

Command Screening Exit (DGRUCD EXEC)

The prolog to the sample command screening exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

Job submissions are screened both by this exit (command is SUBMIT) and by the job screening exit, DGRUJB EXEC.

Note: The variable passed to the exit as *Cmdline*, containing the issued command, is not in a fixed format. The first token is always the command verb. However, the structure, order and content of the remaining tokens varies widely; these tokens are provided for convenience only, no description is provided of their content. There is an exception to this rule: Command cards in command and job files submitted to the Facility through the direct or RJE interface are passed to the exit exactly as submitted.

```
/*                                                              */
/*  Exec Name: DGRUCD                                           */
/*                                                              */
/*          Command Screening Exit for the VM Batch Facility    */
/*                                                              */
/*  Function:                                                   */
/*                                                              */
/*    1. This Exit is called on receipt of all external commands to  */
/*       the VM Batch monitor, before the command is processed. */
/*       It is not called for commands that were generated      */
/*       internally by user exits.                              */
/*                                                              */
/*    2. It audits incoming commands to determine whether they are  */
/*       to be accepted and processed.                         */
/*       In particular, when a command is received from a remote  */
/*       node, its acceptance may be determined by setting the  */
/*       variable 'allow' in the sample code.                   */
/*       Additionally the setting of variable 'allow2' will     */
/*       determine whether commands/jobs issued by task machines  */
/*       will be permitted.                                     */
/*                                                              */
/*    3. Command execution may be affected by setting a return code  */
/*       from this exit:                                        */
/*       exitrc = 0  -->  execute the command if the issuer has the  */
/*                        necessary authority, else reject it with  */
/*                        a message.                           */
/*       exitrc = 1  -->  execute the command unconditionally.  */
/*       exitrc = 2  -->  reject the command, with no message.  */
/*       exitrc = 3  -->  reject the command, with a message.   */
/*                                                              */
/*       Any other value is treated as exitrc = 0.             */
```

Figure 4 (Part 1 of 2). VM Batch Facility Command Screening Exit Prolog

```
/*                                                                  */
/*                                                                  */
/*  Entry conditions:                                               */
/*                                                                  */
/*     The following information is provided to DGRUCD:             */
/*                                                                  */
/*     Parameter   Size (bytes)    Description                      */
/*     _____         */
/*     batalert    8               Id of user receiving alert msgs. */
/*                                 or '?'                           */
/*                                                                  */
/*     No other parameters are provided when batalert is '?'        */
/*                                                                  */
/*     nodeid      8               Id of node command received from.*/
/*     userid      8               Id of user submitting command.   */
/*     authorized  3               Userid an authorized user? (YES/NO)*/
/*     usertype    4               User Type (LOC/NET/RJE)          */
/*     cmdflags    1               Miscellaneous cmd information     */
/*                 1st character: Command origin indicator          */
/*                                 'U' = User originated command     */
/*                                 'T' = Task originated command     */
/*     reserved2   1               Reserved for IBM use             */
/*     cmdline     varies          Parsed command line:             */
/*                                  - Begins with command verb,     */
/*                                    not BATCH                     */
/*                                  - Will be first card after ID   */
/*                                    card for RJE commands         */
/*                                                                  */
/*                                                                  */
/*  Exit conditions:                                                */
/*                                                                  */
/*        Ensure that the return code is set appropriately.         */
/*                                                                  */
```

Figure 4 (Part 2 of 2). VM Batch Facility Command Screening Exit Prolog

Job Screening Exit (DGRUJB EXEC)

The prolog to the sample job screening exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

Note that the job screening exit issues several messages to job submittors. The messages can be removed, changed, or replaced. The messages are not processed by the VM Batch Facility message screening exit (DGRUMSG EXEC).

```
/*                                                             */
/*  Exec Name: DGRUJB                                          */
/*                                                             */
/*          Job Screening Exit for the VM Batch Facility       */
/*                                                             */
/*  Function:                                                  */
/*                                                             */
/*     1. This exit is called at the following stages in the life of */
/*        every job; the parameter 'Called' describes the stage: */
/*         a. Called = 'RECVD'                                  */
/*            When the job is received (SUBMIT processing).     */
/*         b. Called = 'CHJOB'                                  */
/*            If the job is changed (CHJOB processing).         */
/*         c. Called = 'HELD'                                   */
/*            If the job is ready to start but has previously been */
/*              held by this exit. This call is a reminder that the */
/*              job needs to be freed, and provides an opportunity */
/*              to free it.                                     */
/*         d. Called = 'START'                                 */
/*            Before the job is started.                        */
/*         e. Called = 'RESTART'                               */
/*            Before the job is restarted after a system shutdown. */
/*         f. Called = 'JOBEND'                                */
/*            After the job ends (if normal end).              */
/*         g. Called = 'JOBSTEP'                               */
/*            Whenever the monitor receives a BATCH JOBSTEP command */
/*              from a job.                                     */
/*        The DGRIDO exit is called if any abnormal condition arises */
/*        during the life of the job.                          */
```

Figure 5 (Part 1 of 5). VM Batch Facility Job Screening Exit Prolog

```
/*                                                             */
/*     2. It audits all incoming jobs to determine whether they are */
/*        to be accepted and processed.                        */
/*        In particular:                                       */
/*         a. When a job from a remote node is received, its   */
/*            acceptance may be determined by setting the variable */
/*            'allow' in the sample code.                       */
/*         b. When a job from a remote node is accepted or changed, */
/*            or a local user requests or changes an alternate */
/*            userid, then the requested alternate userid MUST be */
/*            confirmed, or an alternative supplied, via the 'ALTID' */
/*            command, otherwise no alternate userid will be set */
/*            by the monitor.                                   */
/*            Confirmation of an alternate userid must be done  */
/*            before the job is STARTed, that is, at the RECVD and */
/*            CHJOB stages.                                     */
/*            The parameter 'Jobflags' contains a flag indicating */
/*            when confirmation is required.                    */
/*            The parameter 'Jobaltuser' contains the alternate */
/*            userid requested by the job submittor.           */
/*         c. When a job is received (Called = 'RECVD'), any LOCJCL */
/*            cards that are included in the job deck are presented */
/*            to this exit in the program stack. These must be  */
/*            removed from the stack prior to leaving the exit. */
/*                                                             */
/*     3. Job execution may be directly affected by setting a return */
/*        code from this exit:                                  */
/*         a. When Called = 'RECVD',                           */
/*            exitrc = 0  -->  accept the job                   */
/*            exitrc = 2  -->  reject the job                   */
/*            exitrc = 3  -->  reject the job with message to job */
/*                             submittor                        */
/*            exitrc = 7  -->  accept the job, but hold it.     */
/*         b. When Called = 'CHJOB',                           */
/*            exitrc = 0  -->  accept the CHJOB                 */
/*            exitrc = 2  -->  cancel the job                   */
/*                             (flushed if recurring)           */
/*            exitrc = 3  -->  cancel the job with message to job */
/*                             submittor and CHJOB issuer       */
/*                             (flushed if recurring)           */
/*            exitrc = 4  -->  reject the CHJOB leaving job unchanged,*/
/*                             with message to CHJOB issuer.    */
/*         c. When Called = 'HELD',                            */
/*            exitrc = 0  -->  leave the job held               */
/*            exitrc = 2  -->  cancel the job                   */
/*                             (rescheduled if recurring)       */
/*            exitrc = 3  -->  cancel the job with message to job */
/*                             submittor                        */
/*                             (rescheduled if recurring)       */
/*            exitrc = 7  -->  leave the job held               */
/*            exitrc = 8  -->  free the job and start it immediately. */
```

Figure 5 (Part 2 of 5). VM Batch Facility Job Screening Exit Prolog

```
/*         d. When Called = 'START',                        */
/*            exitrc = 0  -->  start the job                 */
/*            exitrc = 2  -->  cancel the job                */
/*                       (rescheduled if recurring)          */
/*            exitrc = 3  -->  cancel the job with message to job  */
/*                       submittor                           */
/*                       (rescheduled if recurring)          */
/*            exitrc = 7  -->  hold the job.                 */
/*         e. When Called = 'RESTART',                       */
/*            exitrc = 0  -->  restart the job               */
/*            exitrc = 2  -->  cancel the job                */
/*                       (rescheduled if recurring)          */
/*            exitrc = 3  -->  cancel the job with message to job  */
/*                       submittor                           */
/*                       (rescheduled if recurring)          */
/*         f. When Called = 'JOBEND', the return code is ignored.  */
/*         g. When Called = 'JOBSTEP', the return code is ignored. */
/*                                                           */
/*      Return codes not specified as valid for a particular call  */
/*      are ignored (e.g exitrc = 4 from 'START'); however they do */
/*      generate an error message, and the action for exitrc = 0   */
/*      is performed.                                        */
/*                                                           */
/*     4. Job execution may be indirectly affected by placing */
/*        VM Batch Facility commands on the program stack, for */
/*        execution by the Facility on completion of its current */
/*        activity.                                          */
/*        In particular, the 'ALTID' command is provided for the */
/*        confirmation of alternate userids.                 */
/*        Such stacked commands are subject to the following */
/*        restrictions:                                      */
/*         a. Only a subset of Facility commands are available. */
/*         b. No more than 5 commands may be stacked at each call. */
/*         c. Commands must be in their authorized form, that is, */
/*            every command must explicitly specify the target job. */
/*         d. Commands must not contain the 'BATCH' prefix.  */
/*         e. Commands will be assumed to be authorized.     */
/*         f. Commands will not invoke DGRUCD, DGRUJB or DGRIDO exits */
/*            in their processing.                           */
/*                                                           */
/*  Entry conditions:                                        */
/*                                                           */
/*     The following information is provided to DGRUJB:      */
/*                                                           */
/*     Parameter   Size (bytes)    Description               */
/*     _____   */
/*     Called       7              Reason user exit is called. */
/*                                 (See above) , also '?'.   */
/*                                                           */
/*     No other parameters are provided when Called is '?'   */
/*                                                           */
```

Figure 5 (Part 3 of 5). VM Batch Facility Job Screening Exit Prolog

```
/*      Batalert      8                Id of user receiving alert msgs.   */
/*      Jobnode       8                Id of node job received from.      */
/*      Jobuser       8                Id of user submitting job.         */
/*      Jobname       8                Job name assigned.                 */
/*      Jobclass      8                Job class assigned.                */
/*      Jobacct       8                Job accounting information.        */
/*      Jobdate      10                Date job scheduled to run          */
/*                                       (mm/dd/ccyy or dd.mm.ccyy)       */
/*      Jobpriority   4                Job queueing priority (0..9)       */
/*      Jobwb         4                Job window begin. (hhmm)           */
/*      Jobwe         4                Job window end.   (hhmm/OPEN)      */
/*      Jobaltuser    8                Alternate userid                   */
/*                                     This is the altid as requested by  */
/*                                     the submittor or as set by an      */
/*                                     ALTID or CHJOB command.            */
/*      Jobrecp       8                Recurrence pattern. Set to '*' if  */
/*                                     job does not recur explicitly.     */
/*      Reserved      1                Reserved for IBM use               */
/*      Jobflags      5                Miscellaneous job information      */
/*                    1st character: Alternate userid confirmation        */
/*                                     'C' = confirmed or not required    */
/*                                     'U' = confirmation required        */
/*                    2nd character: Submittor type                       */
/*                                     'L' = Local user                   */
/*                                     'N' = Network user (remote node)   */
/*                                     'R' = RJE device                   */
/*                    3rd character: LLS subjection                       */
/*                                     'L' = LLS subject                  */
/*                                     'I' = Independent of LLS           */
/*                    4th character: Job origin indicator                 */
/*                                     'U' = User originated job          */
/*                                     'T' = Task originated job          */
/*                    5th character: Job recurrence indicator             */
/*                                     'R' = Recurs explicitly by jobrecp */
/*                                     'C' = Recurs implicitly by chain   */
/*                                     'N' = Does not recur               */
/*      Jobsecond     8                Maximum seconds to run job.        */
/*      Jobprint      4                Maximum print records for job.     */
/*      Jobpunch      4                Maximum punch records for job.     */
/*      Jobrestart    3                Job is restartable   (YES/NO)      */
/*      Jobdump       8                Job dump option. (YES/NO/fn)       */
/*                                       (For class AUTOLOG, jobdump will */
/*                                        contain name of job exec, if    */
/*                                        one was defined on SUBMIT)      */
/*      Jobchain     26                Job to be chained to. (* /u.j.<s>) */
```

Figure 5 (Part 4 of 5). VM Batch Facility Job Screening Exit Prolog

```
/*      Jobtask     8                   Task machine that job is assigned  */
/*      Issuser     8                   Userid that issued CHJOB           */
/*      Issnode     8                   Node of userid that issued CHJOB   */
/*      Jobfinalrc  8                   Job's final return code            */
/*                                         (valid for reason JOBEND only   */
/*                                          '*' for all other reasons)     */
/*      Jobstor     8                   Virtual storage (in KB) for job    */
/*                                         (can also be '*' if defaulted)  */
/*      Jobstep     8                   Last recorded jobstep ( '*' if no  */
/*                                         BATCH JOBSTEPs were received ). */
/*                                         In particular, this is '*' and  */
/*                                         Called is 'JOBSTEP' when the    */
/*                                         task machine signals that the   */
/*                                         job is starting.                */
/*                                                                         */
/*      In addition, when Called = 'RECVD', the program stack             */
/*      contains any LOCJCL cards that are present in the job deck.        */
/*                                                                         */
/*  Exit conditions:                                                       */
/*                                                                         */
/*      1. Ensure that the return code is set appropriately.              */
/*      2. Ensure that the stack contains only the required               */
/*         VM Batch commands.                                              */
/*                                                                         */
```

Figure 5 (Part 5 of 5). VM Batch Facility Job Screening Exit Prolog

Job Problem Handling Exit (DGRIDO EXEC)

The prolog to the sample job problem handling exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                            */
/*  Exec Name: DGRIDO                                         */
/*                                                            */
/*           Job Problem Exit for the VM Batch Facility       */
/*                                                            */
/*  Function:                                                 */
/*                                                            */
/*     1. This exit is called when an abnormal condition arises */
/*        during the processing of a job. The parameter 'Called' */
/*        describes the condition:                            */
/*         a. Called = 'STOP'                                 */
/*             When the job is cancelled by a 'STOP JOBS' command. */
/*         b. Called = 'LIMITS'                               */
/*             When the job has exceeded its CPU, PUN or PRT limit. */
/*         c. Called = 'STALL'                                */
/*             When the job has stalled.                      */
/*         d. Called = 'FORCE'                                */
/*             When the job is stopped by a 'FORCE' command against */
/*             the task machine in which it was running.      */
/*         e. Called = 'CANCEL'                               */
/*             When the job is cancelled by a 'CANCEL' command. */
/*         f. Called = 'NSTART'                               */
/*             When the job did not start on its designated task */
/*             machine. This is usually caused by environmental */
/*             factors such as expired passwords. The job is requeued. */
/*         g. Called = 'AUTOERR'                              */
/*             When the job did not start on its designated task */
/*             machine, or a class AUTOLOG job failed to start. This */
/*             is usually caused by job submittor error. The job is */
/*             cancelled. If recurring it is flushed.         */
/*         h. Called = 'RJEALTU'                              */
/*             When a job is about to start, but required confirmation */
/*             of a requested alternate userid has not been given by */
/*             DGRUJB EXEC. The VM Batch monitor has not and will not */
/*             set an alternate userid. At this stage, the task */
/*             machine that will run the job has been autologged, so */
/*             this exit may now set an alternate userid for the job */
/*             if required.                                   */
/*             This exit is also called for this reason when such a */
/*             job is cancelled, and the task machine has been */
/*             autologged again to perform error processing.  */
/*                                                            */
/*         The DGRUJB exit is called to monitor the normal stages of */
/*         a job.                                             */
```

Figure 6 (Part 1 of 4). VM Batch Facility Job Problem Exit Prolog

```
/*                                                            */
/*     2. Job execution may be directly affected by setting a return  */
/*        code from this exit:                                */
/*           -  When Called = 'STOP', the return code is ignored,    */
/*              and the job is cancelled with message to owner.    */
/*              It is rescheduled if recurring.                */
/*           -  When Called = 'LIMITS',                       */
/*              exitrc = 0  -->  cancel the job, with message to owner  */
/*                              (rescheduled if recurring)    */
/*              exitrc = 2  -->  cancel the job with no message   */
/*                              (rescheduled if recurring)    */
/*              exitrc = 3  -->  cancel the job, with message to owner  */
/*                              (rescheduled if recurring)    */
/*              exitrc = 5  -->  let the job continue, and call this  */
/*                              exit if the limits are exceeded.  */
/*              exitrc = 6  -->  let the job continue but suppress  */
/*                              monitoring of limits for this job.  */
/*           -  When Called = 'STALL',                        */
/*              exitrc = 0  -->  if no IUCV paths are open, then cancel  */
/*                              the job with message to owner; else  */
/*                              let it continue and call this  */
/*                              exit if the stall persists.   */
/*                              Cancel will reschedule a recurring job  */
/*              exitrc = 2  -->  cancel the job unconditionally, with  */
/*                              no message to owner          */
/*                              (rescheduled if recurring)    */
/*              exitrc = 3  -->  cancel the job unconditionally, with  */
/*                              message to owner             */
/*                              (rescheduled if recurring)    */
/*              exitrc = 5  -->  let the job continue, and call this  */
/*                              exit if the stall persists.   */
/*              exitrc = 6  -->  let the job continue but suppress  */
/*                              stall monitoring for this job;  */
/*                              including the calling of this exit.  */
/*           -  When Called = 'FORCE', the return code is ignored,   */
/*              and the job is cancelled with message to owner.    */
/*              It is rescheduled if recurring.                */
/*           -  When Called = 'CANCEL', the return code is ignored,  */
/*              and the job is cancelled with message to owner.    */
/*              It is rescheduled if recurring, and was not flushed.  */
/*           -  When Called = 'NSTART',                       */
/*              exitrc = 0  -->  requeue the job (DGRUJB EXEC will  */
/*                              be called again when the job restarts)  */
/*              exitrc = 2  -->  cancel the job with no message   */
/*                              (rescheduled if recurring)    */
/*              exitrc = 3  -->  cancel the job with message to owner.  */
/*                              (rescheduled if recurring)    */
/*           -  When Called = 'AUTOERR', the return code is ignored,  */
/*              and the job is cancelled with message to owner.    */
/*              If recurring it is flushed.                    */
/*           -  When Called = 'RJEALTU', the return code is ignored,  */
/*              and the job proceeds.                          */
/*                                                            */
```

Figure 6 (Part 2 of 4). VM Batch Facility Job Problem Exit Prolog

```
/*      4. Job execution may be indirectly affected by placing    */
/*         VM Batch Facility commands on the program stack, for    */
/*         execution by the Facility on completion of its current  */
/*         activity.                                               */
/*         Such stacked commands are subject to the following      */
/*         restrictions:                                           */
/*          a. Only a subset of Facility commands are available.   */
/*          b. No more than 5 commands may be stacked at each call.*/
/*          c. Commands must be in their authorized form, that is, */
/*             every command must explicitly specify the target job.*/
/*          d. Commands must not contain the 'BATCH' prefix.       */
/*          e. Commands will be assumed to be authorized.          */
/*          f. Commands will not invoke DGRUCD, DGRUJB or DGRIDO exits */
/*             in their processing.                                */
/*                                                                 */
/*  Entry conditions:                                              */
/*                                                                 */
/*      The following information is provided to DGRIDO:           */
/*                                                                 */
/*      Parameter   Size (bytes)    Description                    */
/*      _____   */
/*      Called       7              Reason user exit is called.    */
/*                                  (See above), also '?'.         */
/*                                                                 */
/*      No other parameters are provided when Called is '?'        */
/*                                                                 */
/*      Batalert     8              Id of user receiving alert msgs.*/
/*      Jobnode      8              Id of node job received from.  */
/*      Jobuser      8              Id of user submitting job.     */
/*      Jobname      8              Job name assigned.             */
/*      Jobclass     8              Job class assigned.            */
/*      Jobacct      8              Job accounting information.    */
/*      Jobdate     10              Date job scheduled to run      */
/*                                     (mm/dd/ccyy or dd.mm.ccyy)  */
/*      Jobpriority  4              Job queueing priority (0..9)   */
/*      Jobwb        4              Job window begin. (hhmm)       */
/*      Jobwe        4              Job window end.   (hhmm/OPEN)  */
/*      Jobaltuser   8              Alternate userid               */
/*                                  This is the altid as requested by */
/*                                  the submittor or as set by an  */
/*                                  ALTID or CHJOB command.        */
/*      Jobrecp      8              Recurrence pattern. Set to '*' if */
/*                                  job does not recur explicitly. */
/*      Reserved     1              Reserved for IBM use           */
```

Figure 6 (Part 3 of 4). VM Batch Facility Job Problem Exit Prolog

```
/*     Jobflags     6              Miscellaneous job information     */
/*                  1st character: User type                        */
/*                                 'L' = Local user                 */
/*                                 'N' = Network user (remote node)  */
/*                                 'R' = RJE device                 */
/*                  2nd character: LLS subjection                   */
/*                                 'L' = LLS subject                */
/*                                 'I' = Independent of LLS         */
/*                  3rd character: IUCV Paths at last stall detection */
/*                                 'P' = IUCV paths open            */
/*                                 '*' = No open paths or not relevant*/
/*                  4th character: Limit that is exceeded           */
/*                                 '*' = Not relevant               */
/*                                 '1' = CPU                        */
/*                                 '2' = PRT                        */
/*                                 '4' = PUN                        */
/*                                 If more than one limit is exceeded */
/*                                  then the above values are ORed   */
/*                                  (e.g. 5 = CPU + PUN).            */
/*                  5th character: Job origin indicator             */
/*                                 'U' = User originated job         */
/*                                 'T' = Task originated job         */
/*                  6th character: Job recurrence indicator         */
/*                                 'R' = Recurs explicitly by jobrecp */
/*                                 'C' = Recurs implicitly by chain   */
/*                                 'N' = Does not recur             */
/*     Jobsecond    8              Maximum seconds to run job.       */
/*     Jobprint     4              Maximum print records for job.    */
/*     Jobpunch     4              Maximum punch records for job.    */
/*     Jobrestart   3              Job restarted.    (YES/NO)        */
/*     Jobdump      8              Job dump option.  (YES/NO/fn)     */
/*                                    (For class AUTOLOG, jobdump will */
/*                                     contain name of job exec, if  */
/*                                     one was defined on SUBMIT)    */
/*     Jobchain     26             Job to be chained to. (* /u.j.<s>) */
/*     Jobtask      8              Task machine that job is assigned */
/*     Issuser      8              Userid that issued the command    */
/*                                 giving rise to the condition      */
/*                                 ( e.g. CANCEL). If the condition  */
/*                                 is not in response to a command   */
/*                                 then this field is '*'.           */
/*     Issnode      8              Node of userid that issued command */
/*     Jobstall     8              Length of stall in seconds        */
/*     Jobstep      8              Last recorded user JOBSTEP ( '*'  */
/*                                     if no user JOBSTEPs received)  */
/*                                                                   */
/*  Exit conditions:                                                 */
/*                                                                   */
/*    1. Ensure that the return code is set appropriately.          */
/*    2. Ensure that the stack contains only the required           */
/*       VM Batch commands.                                          */
/*                                                                   */
```

Figure 6 (Part 4 of 4). VM Batch Facility Job Problem Exit Prolog

Message Screening Exit (DGRUMSG EXEC)

The prolog to the sample message screening exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                                 */
/* Exec name: DGRUMSG                                              */
/*                                                                 */
/*          Message Screening Exit for the VM Batch Facility       */
/*                                                                 */
/* Function:                                                       */
/*                                                                 */
/*     1. This Exit is called immediately before a message is to be */
/*        sent to a user by the VM Batch Facility monitor.         */
/*                                                                 */
/*        It is not invoked for messages generated by any of the  */
/*        user exits, nor for messages that do not address a user. */
/*        It is not invoked for messages prefixed by DGRUVM and    */
/*        DGREXE. These messages are generated by the user         */
/*        interface modules which do not run in the monitor        */
/*        virtual machine.                                          */
/*                                                                 */
/*     2. It may audit the message and decide whether to let       */
/*        the VM Batch Facility monitor send the message or not.   */
/*                                                                 */
/*        In particular, if you wish to alter the message text,    */
/*        or the userid that is to receive it, or the method by    */
/*        which it is sent, you must send the message yourself.    */
/*                                                                 */
/*     3. Monitor processing of a message may be affected by       */
/*        setting a return code from this exit:                    */
/*                                                                 */
/*        msgrc = 0  -->  send the original message to the original */
/*                        destination.                             */
/*        msgrc = 1  -->  do not send any message.                 */
/*                                                                 */
/*       Any other value is treated as msgrc = 0.                  */
/*                                                                 */
/*        The original message is reflected to the monitor console */
/*        regardless of the return code setting.                   */
/*                                                                 */
```

Figure 7 (Part 1 of 2). VM Batch Facility Message Screening Exit Prolog

```
/*                                                                  */
/* Entry Conditions:                                                */
/*                                                                  */
/*    The following information is provided to DGRUMSG:             */
/*                                                                  */
/*    Parameter    Size (bytes)    Description                      */
/*    _____  */
/*                                                                  */
/*    alertee      8               ID of user receiving alert messages*/
/*                                 or '?'                           */
/*                                                                  */
/*     No other parameters are provided when alertee is '?'         */
/*                                                                  */
/*    msgno        10              Message number being issued      */
/*    userid       8               Userid to whom the message is    */
/*                                  being sent                      */
/*    nodeid       8               Nodeid of user                   */
/*    usertype     4               Type of user:                    */
/*                                 LOC = local user; the monitor    */
/*                                       communicates using         */
/*                                       CP MSGNOH                   */
/*                                 NET = remote user; the monitor   */
/*                                       communicates using         */
/*                                       SMSG RSCS                   */
/*                                 RJE = RJE user; the monitor      */
/*                                       communicates using         */
/*                                       spool files.               */
/*    msgflags     1               Miscellaneous message flags:     */
/*                 1st character: Command / Job origin indicator:   */
/*                                 If the message is being sent to  */
/*                                 a command issuer then:           */
/*                                  'U' = User originated command   */
/*                                  'T' = Task originated command   */
/*                                 If the message is being sent to  */
/*                                 a job owner then:                */
/*                                  'U' = User originated job       */
/*                                  'T' = Task originated job       */
/*                                 If the message is being sent to  */
/*                                 the alertee then:                */
/*                                  unconditionally set to '*'      */
/*                                                                  */
/*    The text of message to be issued may be pulled from the       */
/*    program stack. It is prefixed by the message number.          */
/*                                                                  */
/*    The message number has the format     DGRxxxnnns             */
/*       where:                                                     */
/*         DGRxxx   is the issuing module name                      */
/*         nnn      is the message number                           */
/*         s        indicates the message type (E-error,           */
/*                  I - informational, W - Warning, etc.)           */
/*                                                                  */
/* Exit Conditions:                                                 */
/*                                                                  */
/*        Ensure that the return code is set appropriately.         */
/*                                                                  */
```

Figure 7 (Part 2 of 2). VM Batch Facility Message Screening Exit Prolog

Job Recurrence Exit (DGRECUR EXEC)

The prolog to the sample job recurrence exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                          */
/*  Exec Name: DGRECUR                                      */
/*                                                          */
/*           Job Recurrence Exit for the VM Batch Facility  */
/*                                                          */
/*  Function:                                               */
/*                                                          */
/*     1. This exit called to handle the scheduling of jobs that  */
/*        repeat automatically according to recurrence patterns   */
/*        defined in this exit.                                    */
/*     2. It is only called in relation to jobs that have been made  */
/*        to recur explicitly by the use of the REPEAT option of the */
/*        BATCH SUBMIT command. It is not called to handle the       */
/*        scheduling of jobs that repeat implicitly because they are  */
/*        chained to a recurring job.                                 */
/*     3. It is not called when a job does not recur automatically,   */
/*        that is, when the job has the reserved recurrence pattern   */
/*        KEEP.                                                       */
/*     4. It is not called to process SUBMIT or CHJOB commands when   */
/*        the REPEAT option on the command specifies a reserved       */
/*        pattern name. The reserved names are:                       */
/*         'HOLD'   indicates a request to suspend recurrence         */
/*         'NOHOLD' indicates a request to resume recurrence          */
/*         'NEVER'  indicates  a job that does not recur              */
/*         'KEEP'   indicates a job that does not recur automatically */
/*        You should not provide any patterns with these names.       */
/*                                                                    */
/*     5. The mechanism available to this exit for scheduling an      */
/*        occurrence of a job is to place a VM Batch Facility CHJOB   */
/*        command on the program stack, specifying the DATE option.   */
/*        Other options may also be specified, but DATE is required.  */
/*        Further requirements concerning stacked commands are        */
/*        described later in this prologue.                           */
/*                                                                    */
```

Figure 8 (Part 1 of 4). VM Batch Facility Job Recurrence Exit Prolog

```
/*      6. The exit is called by the Facility monitor at the       */
/*         following times:                                        */
/*           a. At monitor start time, and whenever the installation */
/*              exits are refreshed by a BATCH RELOAD command, the exit */
/*              is called with action = '?' to identify itself on the */
/*              monitor console.                                   */
/*           b. On receipt of every SUBMIT or CHJOB command specifying */
/*              a non reserved recurrence pattern, the exit is called */
/*              with action = 'ACCEPT' to verify the pattern name.  */
/*           c. On completion (normal or abnormal) of each instance of */
/*              an explicity recurring job, the exit is called      */
/*              with action = 'NEXT' to schedule the next occurrence */
/*              of the job.                                         */
/*              Failure to schedule the next occurrence will result in */
/*              a message to the Facility alertee, and the job will  */
/*              need manual rescheduling.                          */
/*           d. If an explicitly recurring job is prevented from    */
/*              running by a suspension of recurrence, and the job  */
/*              becomes eligible to run then the exit is called with */
/*              action = 'SKIP' to schedule the next occurrence.    */
/*              Failure to schedule the next occurrence will result in */
/*              a message to the Facility alertee, and the job will  */
/*              need manual rescheduling.                          */
/*           e. Before each new instance of an explicitly recurring job */
/*              is started, the exit is called with action = 'CONFIRM' */
/*              to verify that the actual start time and date of the */
/*              job remains consistent with the pattern definition, */
/*              since the actual time that a job gets dispatched may */
/*              differ from the scheduled start time and date.      */
/*              If the exit prevents the start of the job, then it must */
/*              schedule the next occurrence. Failure to do so will  */
/*              result in a message to the alertee, and the job will */
/*              need manual rescheduling.                          */
/*           f. On receipt of a BATCH QUERY REPEATS command the exit is */
/*              called with action = 'QUERY'. The exit must reply   */
/*              directly to the user indicating which repeat patterns */
/*              are available.                                     */
/*                                                                 */
/*      7. Job execution may be directly affected by setting a return */
/*         code from this exit:                                    */
/*           a. When Action = '?',                                 */
/*              the return code is ignored.                        */
/*           b. When Action = 'ACCEPT',                            */
/*              exitrc = 0  -->  accept the job/CHJOB.             */
/*              exitrc = 2  -->  reject the job/CHJOB, no message to */
/*                               owner.                            */
/*              exitrc = 3  -->  reject the job/CHJOB, with message to */
/*                               owner.                            */
```

Figure 8 (Part 2 of 4). VM Batch Facility Job Recurrence Exit Prolog

```
/*          c. When Action = 'CONFIRM',                        */
/*             exitrc = 0  -->  allow the job to run.          */
/*             exitrc = 2  -->  do not run the job now,        */
/*                              next scheduled date has been stacked. */
/*          d. When Action = 'NEXT',                           */
/*             the return code is ignored.                     */
/*          e. When Action = 'SKIP',                           */
/*             the return code is ignored.                     */
/*          f. When Action = 'QUERY',                          */
/*             the return code is ignored.                     */
/*                                                             */
/*       Return codes not specified as valid for a particular call */
/*       are ignored (e.g return code = 1 when action = 'CONFIRM'); */
/*       however they do generate an error message, and the action */
/*       for exitrc = 0 is performed.                          */
/*                                                             */
/*       8. Job execution may be indirectly affected by placing */
/*          VM Batch Facility commands on the program stack, for */
/*          execution by the Facility on completion of its current */
/*          activity.                                          */
/*          In particular, the 'CHJOB' command is stacked to schedule */
/*          the next occurrence date for each recurring job.   */
/*          Such stacked commands are subject to the following */
/*          restrictions:                                      */
/*          a. Only a subset of Facility commands are available. */
/*          b. No more than 5 commands may be stacked at each call. */
/*          c. Commands must be in their authorized form, that is, */
/*             every command must explicitly specify the target job. */
/*          d. Commands must not contain the 'BATCH' prefix.   */
/*          e. Commands will be assumed to be authorized.      */
/*          f. Commands will not invoke DGRUCD, DGRUJB, DGRIDO, or */
/*             DGRECUR in their processing.                    */
/*                                                             */
/*  Entry Conditions:                                          */
/*                                                             */
/*     The following information is provided to DGRECUR.       */
/*         All characters are in UPPER case.                   */
/*                                                             */
/*                Size                                         */
/*     Parameter  (bytes)  Description                         */
/*     _____ */
/*     Action       7      - the action expected from the exit: */
/*                            '?' - identify this exit.        */
/*                          'QUERY' - tell user of available patterns */
/*                          'ACCEPT' - verify a requested pattern */
/*                         'CONFIRM' - confirm that a job may run now */
/*                            'NEXT' - schedule the next run of a job */
/*                            'SKIP' - handle a job missing its run */
/*     Batalert     8      - userid of facility alertee.       */
/*                                                             */
/*     No other parameters are provided when Action is '?'     */
```

Figure 8 (Part 3 of 4). VM Batch Facility Job Recurrence Exit Prolog

```
/*                                                             */
/*      Jobnode     8      - nodeid of job owner or Query issuer  */
/*      Jobuser     8      - userid of job owner or Query issuer  */
/*                                                             */
/*All following parameters except Utype are '*' when Action is 'QUERY'*/
/*                                                             */
/*      Jobname     8      - name of job.                      */
/*      Jobclass    8      - class job is to run under.        */
/*      Jobacct     8      - job account number.               */
/*      Jobsdate   10      - scheduled date for occurrence.    */
/*                           (mm/dd/yy, mm/dd/ccyy, dd.mm.yy,  */
/*                            dd.mm.ccyy)                      */
/*                           This pertains to this occurence when */
/*                           action = 'CONFIRM','NEXT' or 'SKIP'. */
/*                           This pertains to the next occurence when */
/*                           action = 'ACCEPT'; except if this is a */
/*                           CHJOB command and an instance of the job */
/*                           is currently running.            */
/*      Jobpriority 4      - job priority.                     */
/*      Jobwb       4      - time scheduled start window opens. */
/*      Jobwe       4      - time scheduled start window closes. */
/*      Jobaltuser  8      - alternate userid.                 */
/*                           This is the altid as requested by the */
/*                           submittor or as set by an ALTID or CHJOB */
/*                           command.                          */
/*      Jobrecp     8      - recurrence pattern name.          */
/*      Reserved    1      Reserved for IBM use                */
/*      Utype       1      - user type.                        */
/*                           'L' = Local user                  */
/*                           'N' = Network user (remote node)  */
/*                           'R' = RJE device (talk in spool files) */
/*      Jobqtime   18      - date and time job was (re)queued  */
/*                           (mm/dd/ccyyhh:mm:ss)              */
/*      Jobrstar   18      - date and time this occurence started. */
/*                           (mm/dd/ccyyhh:mm:ss)              */
/*                           This is '*' when action = 'ACCEPT' or */
/*                           'CONFIRM' or 'SKIP'.              */
/*      Jobrend    18      - date and time this occurence ended */
/*                           (mm/dd/ccyyhh:mm:ss)              */
/*                           excluding error processing, if any. */
/*                           This is '*' when action = 'ACCEPT' or */
/*                           'CONFIRM' or 'SKIP'.              */
/*      Jobrc       8      - return code set by job on last run. */
/*                           This is '*' if never run or last run */
/*                           terminated abnormally.            */
/*      Jobrnum     4      - count of successful job starts.   */
/*                                                             */
/*                                                             */
/*   Exit Conditions:                                          */
/*                                                             */
/*      1. Ensure that the return code is set appropriately.  */
/*      2. Ensure that the stack contains only the required   */
/*         VM Batch commands.                                  */
/*                                                             */
```

Figure 8 (Part 4 of 4). VM Batch Facility Job Recurrence Exit Prolog

User Interface Exit (DGREVR EXEC)

The prolog to the sample user interface exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                          */
/*  Exec Name: DGREVR                                       */
/*                                                          */
/*       User Interface Exit for the VM Batch Facility      */
/*                                                          */
/*  Function:                                               */
/*                                                          */
/*    1. This Exit is called by the user interface of the VM Batch  */
/*       Facility, DGRBATCH EXEC, which runs in the virtual machine  */
/*       on which a VM Batch command is invoked.            */
/*                                                          */
/*    2. It is called for all VM Batch commands except JOBSTEP and  */
/*       ADDRESS; however, it is only able to affect the subsequent  */
/*       processing of SUBMIT commands.                     */
/*                                                          */
/*    3. This exit may supply DEFAULT options when processing a  */
/*       SUBMIT command by placing them in the program stack:  */
/*           - Only OPTIONS of the command may be specified.  */
/*           - They are coded in the same syntax as when coded on a  */
/*             SUBMIT command line.                         */
/*           - These options are overridden by command line options.  */
/*           - Multiple options may be supplied.           */
/*           - Multiple stack lines may be used.           */
/*                                                          */
/*    4. This exit may supply installation defined job control cards  */
/*       (LOCJCL cards) to be included in a job deck when processing  */
/*       a SUBMIT command, by placing them in the program stack.  */
/*       These LOCJCL cards are subsequently made available to the  */
/*       addressed monitor's job screening exit (DGRUJB EXEC) when  */
/*       the job is received, and to the task control installation  */
/*       exit (DGROOLY EXEC) when the job is executed.      */
/*           - The format of a LOCJCL card is              */
/*                   LOCJCL text                            */
/*             where 'text' may not exceed 73 characters.   */
/*           - Up to 3 LOCJCL cards may be supplied.       */
/*           - Each card must be stacked separately.       */
/*                                                          */
```

Figure 9 (Part 1 of 2). VM Batch Facility User Interface Exit Prolog

```
/*     5. No VM BATCH Facility commands should be issued from within */
/*        this installation-exit.                                    */
/*                                                                   */
/*  Entry conditions:                                                */
/*                                                                   */
/*     The following arguments are provided to this Exit:            */
/*                                                                   */
/*     Arguments                   Description                       */
/*     _____  */
/*     monitor                Id of addressed VM Batch monitor.      */
/*     mnode                  Node of addressed monitor.             */
/*     cmdverb                VM Batch command name, unabbreviated.  */
/*                               e.g. SUBMIT or CANCEL               */
/*     cmdLine                Rest of the command line, as entered.  */
/*                                                                   */
/*     Example:   If the issuer entered                             */
/*                BATCH SUB JOBA EXEC ( CLASS K DA HOLD              */
/*                then                                               */
/*                    Cmdverb contains    SUBMIT                     */
/*                    Cmdline contains    JOBA EXEC ( CLASS K DA HOLD */
/*                                                                   */
/*  Exit conditions:                                                 */
/*                                                                   */
/*     Any material in the program stack is deemed to contain default */
/*     options and LOCJCL cards to be included with a SUBMIT.        */
/*                                                                   */
/*     The return code is ignored.                                  */
/*                                                                   */
```

Figure 9 (Part 2 of 2). VM Batch Facility User Interface Exit Prolog

Task Control Exit (DGROOLY EXEC)

The prolog to the sample task control exit supplied with the VM Batch Facility is reprinted below. It explains the function of the exit and describes the VM Batch Facility interface to the exit.

Upon receiving control, the exec parses the input parameters into a set of variables available to the exec. These variable names are listed under the heading 'parameter' in the prolog listing.

```
/*                                                          */
/*  Exec Name: DGROOLY                                      */
/*                                                          */
/*        Task Control Exit for the VM Batch Facility       */
/*                                                          */
/*  Function:                                               */
/*                                                          */
/*    1. This Exit is called by the task control program of the   */
/*       VM Batch Facility (DGRTASK EXEC), which runs in the virtual */
/*       machine on which a VM Batch job is executed.       */
/*                                                          */
/*    2. It is called whenever a VM Batch task machine is   */
/*       initialized to execute a job on behalf of its owning */
/*       VM Batch monitor. This may be for the initial start of a */
/*       job, or a restart of an interrupted job, or for error */
/*       processing. It is called before any processing of the */
/*       job commences, and in the case of an initial start, before */
/*       the job has been read onto disk.                   */
/*                                                          */
/*    3. When called for an initial job start, this exit is supplied */
/*       with any installation defined job control cards (LOCJCL */
/*       cards) that are present in the job deck. These cards are */
/*       available in the program stack.                    */
/*                                                          */
/*    4. Care should be taken in the coding of this exit:   */
/*       a. Since the job has not yet been loaded from the reader */
/*          (at initial start), the reader should not be interfered */
/*          with.                                           */
/*       b. No VM Batch Facility commands, including JOBSTEP, should */
/*          be issued from within this exit.                */
/*       c. Spooling and tagging of the virtual console, printer and */
/*          punch should not be changed.                    */
/*       d. Excessive processing should be avoided.         */
/*                                                          */
```

Figure 10 (Part 1 of 2). VM Batch Facility Task Control Exit Prolog

```
/*      5. Console spooling is started before this exit is called, so  */
/*         any messages generated by this exit appear on the job's     */
/*         console log.                                                 */
/*                                                                      */
/*      6. Job execution is not affected by setting a return code from */
/*         this exit.                                                   */
/*                                                                      */
/*      7. The program stack is cleared by DGRTASK on return from      */
/*         this exit.                                                   */
/*                                                                      */
/*                                                                      */
/*  Entry conditions:                                                   */
/*                                                                      */
/*      The following information is provided to this Exit:            */
/*                                                                      */
/*      Parameter                   Description                         */
/*      _____ */
/*      Reason                      START, RESTART or ERROR.            */
/*      Jobowner                    Userid of job owner.                */
/*      Jobnode                     Node on which job owner resides.    */
/*      Jobname                     Name (ID) of the job.               */
/*      Monid                       ID of the monitor owning this       */
/*                                  task machine.                       */
/*                                                                      */
/*      In addition, when called for reason = START, the program       */
/*      stack contains any LOCJCL cards that were included in the       */
/*      job deck.                                                       */
/*                                                                      */
```

Figure 10 (Part 2 of 2). VM Batch Facility Task Control Exit Prolog

Chapter 6. VMBATCH DEFAULTS File

In the same way that the CONTROL FILE provides configuration parameters to a VM Batch Facility monitor within its virtual machine, the VMBATCH DEFAULTS file may be used to provide configuration parameters to those components of the VM Batch Facility that do not necessarily run in the monitor virtual machine; for example the task and user interfaces.

This is achieved by placing the VMBATCH DEFAULTS file on a disk available to all VM Batch Facility users, task machines and monitors; for example, the MAINT 19E disk.

However, while the CONTROL FILE settings cannot be overridden by a monitor, the VMBATCH DEFAULTS settings can easily be overridden by a user creating a private copy. The VMBATCH DEFAULTS file is therefore only a tool of convenience.

Use of the VMBATCH DEFAULTS file is optional.

It can be used very effectively within an installation to:

- Provide a default destination for all VM Batch Facility commands issued by users, thus avoiding repetitive use of the BATCH ADDRESS command.

- Specify that logon passwords are mandatory for class AUTOLOG jobs submitted by users, and CP AUTOLOG commands issued by VM Batch Facility monitors.

- Specify that logon passwords are not required at all for class AUTOLOG jobs and CP AUTOLOG commands.

- Specify that the user interface module, DGRUVM, be loaded as a CMS nucleus extension in the user's virtual machine.

Creating the VMBATCH DEFAULTS File

Create a VMBATCH DEFAULTS File according to the following general specifications:

- It is a CMS file with *filename* 'VMBATCH', *filetype* 'DEFAULTS' and has a *filemode* number of 2 to allow common access.

- Each statement is of the format 'keyword = value'.

  These are detailed below.

- Only one statement may be placed on each line of the file.

- A line starting with an asterisk (*) is treated as a comment.

- If duplicate statements are encountered then the last statement is used.

- Place the file on an appropriate common disk, where it may be accessed by any VM Batch Facility users, task machines and monitors that will refer to it.

Providing a Default Command Destination

When a user wishes to direct VM Batch jobs and commands to a VM Batch Facility monitor that is not named BATCH or does not reside on the VM system that the user is logged on to, it is necessary for the user to first enter the BATCH ADDRESS command to direct jobs to the desired monitor.

When a VMBATCH DEFAULTS file is accessible to the user, the destination defined in the VMBATCH DEFAULTS file becomes the default for that user; so the user will only need to enter a BATCH ADDRESS command when jobs are to be directed to monitors other than the one specified in the VMBATCH DEFAULTS file.

A destination consists of a monitor name and a node name. Either, both or neither may be specified in the VMBATCH DEFAULTS file.

```
┌─ Format ────────────────────────────────────────────────────────┐
│                                                                   │
│  ►►──MONITOR = monitorid──►◄                                      │
│                                                                   │
│                                                                   │
│  ►►──NODE = nodeid──►◄                                            │
│                                                                   │
└───────────────────────────────────────────────────────────────────┘
```

monitorid
   This is the name of the VM Batch monitor that is to be the default. This entry is only required if the monitor name is not BATCH.

nodeid
   This is the name of the node on which the default monitor resides. This entry is only required if the node on which the default monitor resides is not the same node as that the users are logged on to.

Discarding or Insisting on Logon Passwords

The VM Batch Facility does not insist that a logon password be available when it issues a CP AUTOLOG command. If one is available then it is used, otherwise the command is issued without a password.

For example, when a user submits a CLASS AUTOLOG job and does not provide a password on the command line, the VM Batch Facility prompts the user to provide one. However, a null reply is acceptable, so if no password is provided then no password will be used when the monitor comes to issue the CP AUTOLOG command.

At some installations, a password is mandatory. If a user on such a system gives a null reply to the password prompt, then the monitor goes into a CP READ after issuing the CP AUTOLOG command, effectively crippling it.

On the other hand, installations that have included the CP password suppression facility, do not require logon passwords when issuing CP AUTOLOG commands. If a user on such a system does supply a password for a class AUTOLOG job, the CP AUTOLOG command subsequently issued by the monitor fails with the bad password. If this occurs often enough, CP disables the AUTOLOG command for the monitor, effectively crippling it.

If passwords are mandatory at your installation, then you should prevent possible monitor malfunction by using the VMBATCH DEFAULTS file to insist that logon passwords are provided whenever a function involving the CP AUTOLOG command is invoked. Similarly, if passwords are not required at your installation, then you should use the VMBATCH DEFAULTS file to prevent any logon passwords being used by a monitor.

Specifying that logon passwords are mandatory

Specify that logon passwords are mandatory for a monitor and its users by providing the following line in the VMBATCH DEFAULTS file:

```
  ┌─ Format ───────────────────────────────────────────────────┐
  │                                                             │
  │  ►►──AUTOPASS = YES──►◄                                      │
  │                                                             │
  └─────────────────────────────────────────────────────────────┘
```

When a facility is so configured **to insist** on passwords:

- Users that have access to the VMBATCH DEFAULTS file will not be able to submit CLASS AUTOLOG jobs with a null password.

- Monitors that have access to the VMBATCH DEFAULTS file will ensure that a non null password has been provided for every CLASS AUTOLOG job accepted.

- Monitors that have access to the VMBATCH DEFAULTS file will ignore any TASK statement in the CONTROL FILE if it specifies a null password for the task machine.

- '*NOPASS*', the null password indicator, is not accepted as a logon password.

Note: Do not specify that passwords are mandatory if you have enabled the CP password suppression facility.

Specifying that logon passwords be discarded

Specify that logon passwords are not required by providing the following line in the VMBATCH DEFAULTS file:

```
┌─ Format ─────────────────────────────────────────────────────────┐
│  ►►──AUTOPASS = NO──►◄                                             │
└───────────────────────────────────────────────────────────────────┘
```

When a facility is so configured **to discard** passwords:

- Users that have access to the VMBATCH DEFAULTS file are not prompted for a password when submitting class AUTOLOG jobs.

- Monitors that have access to the VMBATCH DEFAULTS file will ignore any passwords provided for Class AUTOLOG jobs.

- Monitors that have access to the VMBATCH DEFAULTS file will not use passwords when AUTOLOGing task machines, even when they are provided.

- Monitors that have access to the VMBATCH DEFAULTS file will reject all BATCH PASSWORD commands.

Note: While users may override any VMBATCH DEFAULTS specification of insisting or discarding passwords, the VM Batch Facility monitors will not. Such overriding will only cause user inconvenience, and not monitor malfunction.

Specifying an automatic NUCXLOAD of DGRUVM MODULE

The DGRUVM MODULE is used to communicate to the VM Batch Facility monitor from user and task virtual machines.

Whenever a VM Batch Facility command is issued in a user virtual machine, DGRUVM MODULE may be loaded in the CMS user area at X'20000', which can conflict with user programs executing at this address.

You may specify that DGRUVM MODULE not load into the CMS user area, but rather be NUCXLOADed by CMS, by providing the following line in the VMBATCH DEFAULTS file:

```
┌─ Format ──────────────────────────────────────────────┐
│                 ┌─NUCXUVM = NO──┐                      │
│        ►►───────┤               ├───────►◄             │
│                 └─NUCXUVM = YES─┘                      │
└────────────────────────────────────────────────────────┘
```
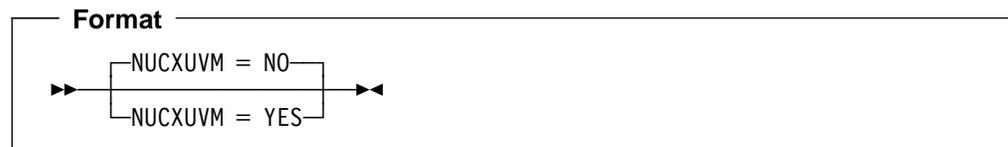
NO

Specify **NO** to allow the DGRUVM MODULE to load in the CMS user area at X'20000'.

This is the default.

YES

Specify **YES** to have the DGRUVM MODULE loaded as a CMS nucleus extension.

Note that the VM Batch Facility will not NUCXDROP the module on completion of a command.

DGRUVM MODULE is always NUCXLOADed when called in a VM Batch Facility task machine.

Sample VMBATCH DEFAULTS file.

Following is an example of a VMBATCH DEFAULTS File. This VMBATCH DEFAULTS file specifies the following:

- Users who have not explicitly addressed another monitor will have their jobs submitted to the monitor named BATCH11 at node NODEAA.

- AUTOLOG passwords will be discarded.

- The VM Batch Facility communications module DGRUVM MODULE will be NUCXLOADed, if it is not already, before it is used in a submittor's virtual machine.

```
┌─ Sample VMBATCH DEFAULTS file ────────────────────────────┐
│                                                           │
│ * VMBATCH DEFAULTS V2.2                                    │
│ MONITOR = BATCH11                                         │
│ NODE = NODEAA                                             │
│ AUTOPASS = NO                                             │
│ NUCXUVM = YES                                             │
│                                                           │
└───────────────────────────────────────────────────────────┘
```

Usage Notes:

1. The VMBATCH DEFAULTS file has no effect on VM Batch Facility commands issued via the Direct or RJE interface.

2. Modifications to a VMBATCH DEFAULTS file take effect at a monitor at startup, or when the BATCH CONTROL SET or BATCH RELOAD commands are issued.

3. If any component of the VM BATCH Facility detects an error in a VMBATCH DEFAULTS file, then a message is issued identifying the statement in error, but processing continues and the statement is ignored. If this occurrs while processing a command, then the VM Batch Facility user interface provides a return code of 4.

4. When permitting spawned jobs on a monitor, be sure to make any VMBATCH DEFAULTS file available to the monitor's task machines.

Chapter 7. Multiple VM Batch Facilities

You are not restricted to running only one VM Batch Facility monitor. Multiple monitors may be set up on a single VM system as required.

You are not restricted to running jobs on local monitors. If you have RSCS links to other VM systems running VM Batch Facility monitors, jobs may be directed to these remote monitors as required.

Users direct their jobs and commands to the monitor of their choice, using the BATCH ADDRESS command.

Every monitor remains a completely separate and individual entity. Monitors cannot share jobs or task machines and do not communicate with each other in any way.

Restriction of access to a monitor is provided by the monitor's exits.

Configuring Multiple VM Batch Facilities

The previous chapters have discussed the installation and operation of a single VM Batch Facility. The software material installed may be divided into two sections. The first consists of user interface modules available to all users of the Facility; the second is for the exclusive use of the Facility monitor. This concept is carried through to the configuration of multiple VM Batch Facilities.

It is not necessary to duplicate the user interface material. It need only be made available to all users, no matter which monitor they address, or which node they run on.

Each monitor, however, is a duplicate installation. Each has its own virtual machine, its own suite of task machines, and its own exits and Control File, and so on.

The following diagram shows a simple single monitor, single node configuration:

User **User** **User**

Task Machine

Monitor Machine BATCH1

Task Machine

Monitor Machine Mini-disks

Task Machine
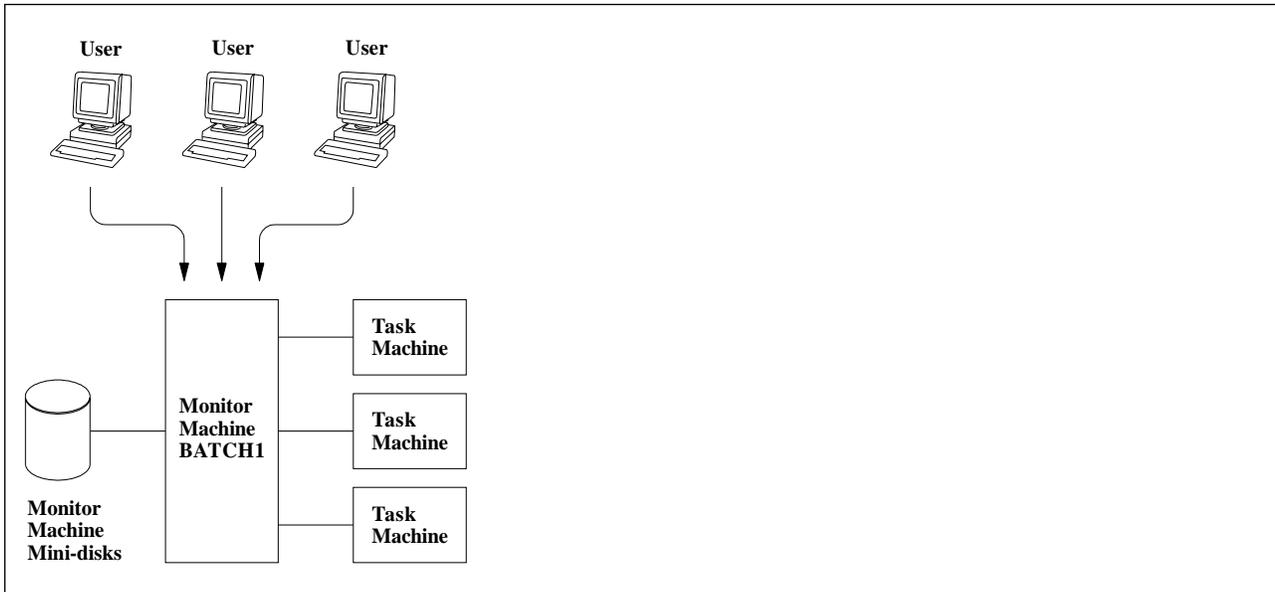
All users in this configuration submit their jobs to monitor BATCH1.

The following diagram shows a simple dual monitor, single node configuration:

User 1 **User 2** **User 3**

Task Machine

Monitor Machine BATCH1

Task Machine

Monitor Machine Mini-disks

Task Machine

Task Machine

Monitor Machine BATCH2

Task Machine

Monitor Machine Mini-disks

Task Machine
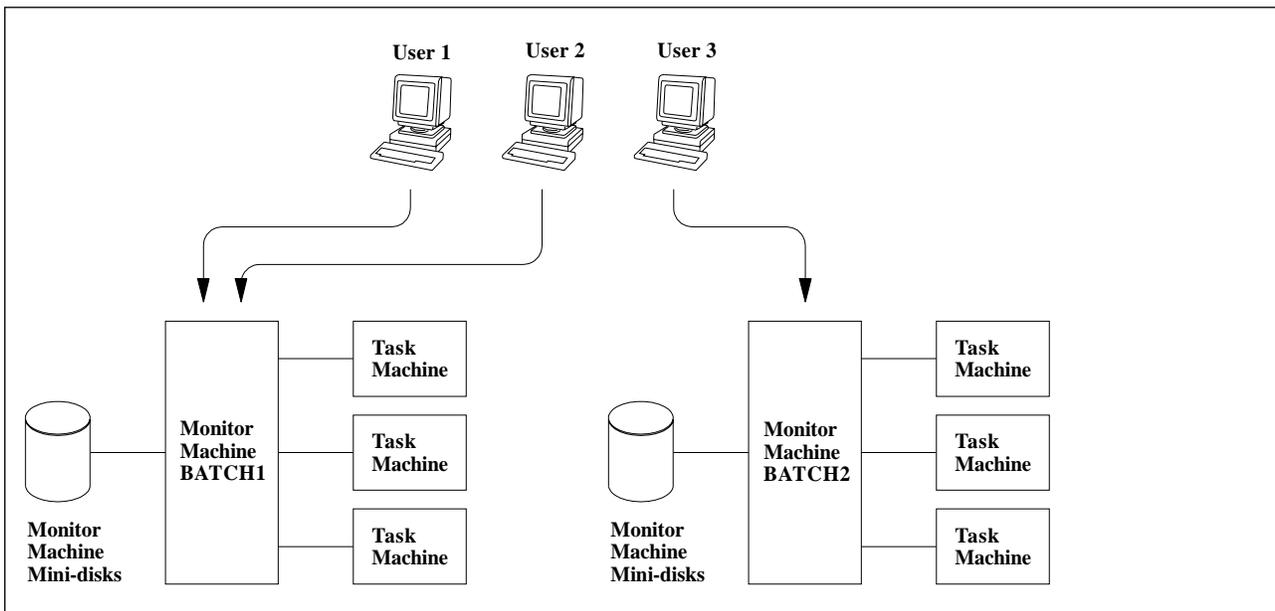
USER1 and USER2 are addressing jobs to monitor BATCH1, while USER3 is addressing jobs to monitor BATCH2.

The following diagram shows a more complex configuration. There are two monitors on one node and a third on another node.
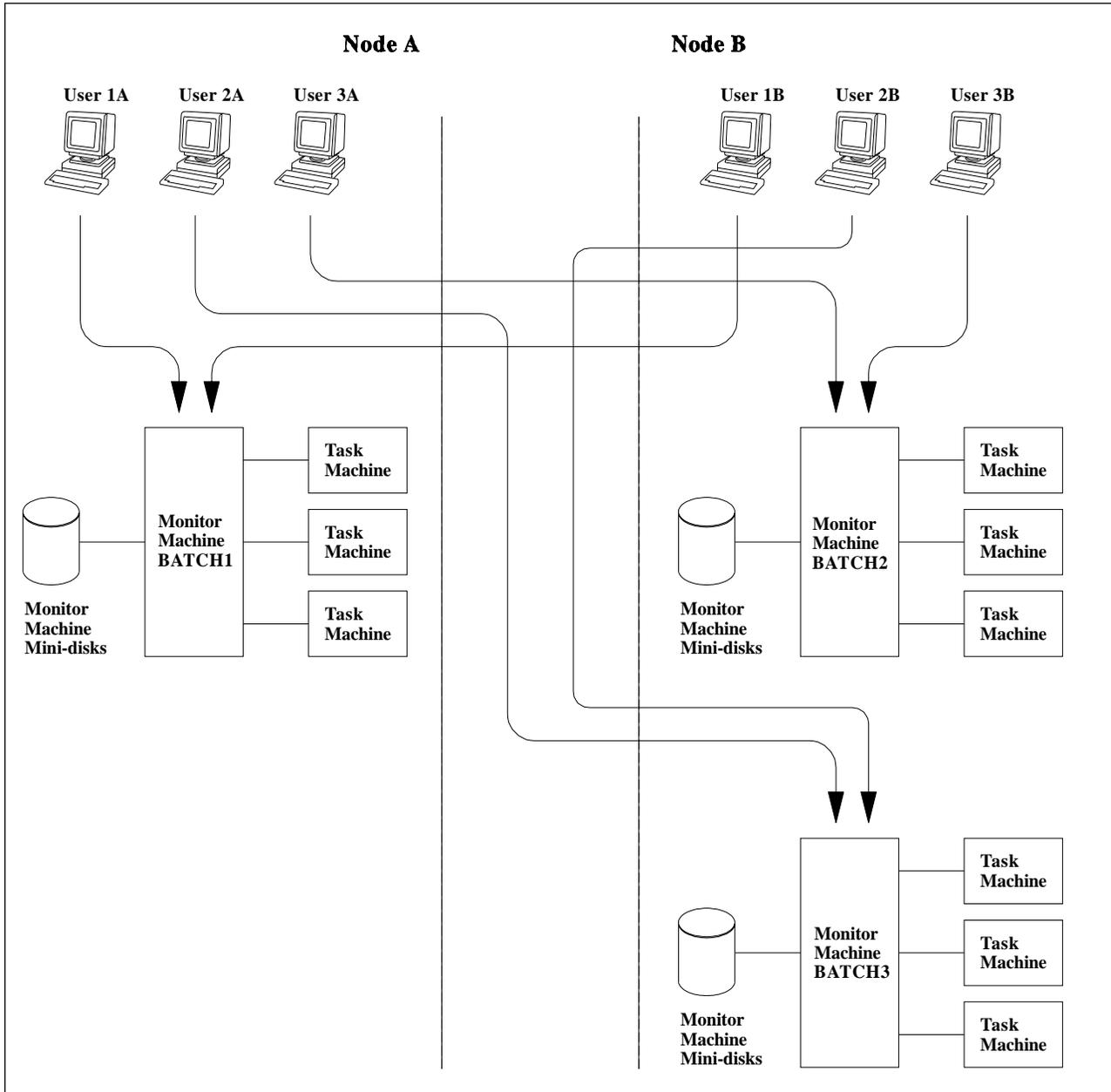
User 1A is addressing jobs to monitor BATCH1 on Node A.
User 2A is addressing jobs to monitor BATCH3 on Node B.
User 3A is addressing jobs to monitor BATCH2 on Node B.
User 1B is addressing jobs to monitor BATCH1 on Node A.
User 2B is addressing jobs to monitor BATCH3 on Node B.
User 3B is addressing jobs to monitor BATCH2 on Node B.

Special Considerations for Multiple VM Batch Facilities

When configuring and using multiple VM Batch monitors, the following items should be taken into account.

- All participating monitors and users must use the same software level of the VM Batch Facility, and that software level must be Version 2 Release 1 or later. Further software constraints may apply. These are described on page 10.

- Each participating monitor machine should be separate from every other. In particular, each monitor must have its own set of task machines, its own Control File, and its own minidisks as if each were a separate installation of the Facility.

  A monitor machine may have any valid user ID.

  Only perform the product installation process to prepare the primary monitor on any one node. Create a secondary monitor by manually preparing the monitor and task virtual machines, and their minidisks, in the image of the first. Sample CP directory entries are provided on page 15.

  Each monitor must be customized separately.

- All the task machines belonging to a monitor must be on the same node as the monitor.

- Security of access to a monitor is controlled by the exits DGRUCD EXEC and DGRUJB EXEC. As shipped, these exits will permit jobs and commands from all local node users, and deny access to remote users. It is your responsibility to tailor these exits to your own security requirements.

- When exits are tailored to permit remote users, remember to configure the exits to provide alternate user IDs for remote users.

- When remote users are permitted to a monitor, the setting of EMSG in the RSCS virtual machine on the remote node should be set to TEXT to allow proper formatting of messages from the monitor to the remote user.

- A user directs jobs and commands to a particular monitor using the BATCH ADDRESS command. Once directed, all commands and jobs will be submitted to that monitor until redirected by another BATCH ADDRESS command. The identity of the chosen monitor and node is saved across sessions in LASTING GLOBALV. The default direction, where no BATCH ADDRESS command has been issued, is to a monitor named BATCH on the user's node.

  You can provide a different default monitor address for your users by creating a VMBATCH DEFAULTS file. See Chapter 6, "VMBATCH DEFAULTS File" on page 159 for more information.

- All commands are available to remote users. In particular, a remote user may also be an administrator of a VM BATCH Facility. A remote user may not be a Facility alertee.

Chapter 8. Messages

All messages issued by the VM Batch Facility have a 10-character identifier that contains the following:

- The first three characters are always DGR, the VM Batch Facility subsystem prefix.

- The fourth through sixth characters identify the VM Batch Facility component issuing the message.

- The seventh through ninth characters are the message number.

- The tenth character is a code for the severity of the condition the message is reporting.

The messages in this section are arranged in two groups:

- Messages with numbers less than 500 are listed in the first group, and arranged in sequence of component ID and message number.

- Messages with numbers of 500 and greater are common to more than one VM Batch Facility component. These messages are listed in the section "Messages Numbered 500 and Higher" on page 224, and are arranged in message number sequence.

VM Batch Facility Severity Codes

The severity codes for all VM Batch Facility messages use the following convention:

Severity Code Meaning

I   No error. This is an informational message only.

W   Indefinite error. A condition has arisen which may or may not constitute an error.

E   Definite error. A function did not complete.

Messages Numbered Less Than 500

The messages in this section are uniquely identified by the combination of a three character code for the VM Batch Facility component issuing the message, and message number. Look up the message using the component code and message number together – the fourth through ninth characters if the message ID.

DGRBAT001E You do not have access to the ISPF facility. If ISPF is installed on your system, link and access the ISPF product minidisk and then retry BATCH MENU. If ISPF is not installed on your system, then you must use the VM Batch Facility command interface for VM Batch Facility functions.

Explanation: You entered BATCH MENU to start the VM Batch Facility panels. The VM Batch Facility was unable to present the panels because your user ID does not have access to the minidisks on which the Interactive System Productivity Facility (ISPF) is stored.

VM Batch Facility Action: The VM Batch Facility was unable to start its panel interface for you.   It is ready for another command.

User Response: If you can, enter the necessary LINK and ACCESS commands to get access to ISPF.   Otherwise, contact your system administrator for help.

You can continue to make use of the VM Batch Facility by entering VM Batch Facility commands.

DGRBAT5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRCMD001E Not permitted to use VM Batch Facility authorized user commands

Explanation: You entered a VM Batch Facility authorized user command, but you have not been identified as an authorized user.   If you are not an authorized user, you can use only the SUBMIT, PASSWORD, QUERY, and JOBSTEP commands, and the nonprivileged forms of the CHJOB, STATUS, and CANCEL commands.

VM Batch Facility Action: Command not performed.   The VM Batch Facility is ready for another command.

User Response: Refer to the *VM Batch Facility User's Guide* for descriptions of the commands you can use.

DGRCMD002E You entered an unknown VM Batch Facility command

Explanation: You entered a command that the VM Batch Facility has not recognized.

VM Batch Facility Action: Command not performed.   The VM Batch Facility is ready for another command.

User Response: Check to see that you used the correct command and that you spelled it right.   Enter the command again if you were mistaken.

DGRCMD003E You entered an incorrect option in a VM Batch Facility command

Explanation: You entered a VM Batch Facility command that included one or more incorrect options.   The problem may be that the option was misspelled, was abbreviated incorrectly, does not belong with the command, or was entered with an invalid value following it.

VM Batch Facility Action: Command not performed.   The VM Batch Facility is ready for another command.

User Response: Check the command format and reenter the command.   To get help with VM Batch Facility command formats at your terminal, enter HELP BATCH.

DGRCMD004E Job *userid nodeid jobid* **not found**

Explanation: You entered a VM Batch Facility command specifying the job named in the message.   However, the program could not find the job.   The problem has occurred for one of these reasons:

1. The program has finished or canceled the job, and the job owner has already seen the final status of the job.   The program erases status records after the job owner has seen the job's final status.

2. The job ID or user ID was misspelled or incorrect.

3. The job has not been submitted to the VM Batch Facility.

4. The VM Batch Facility has had to reuse the control block that recorded information about this job.   The program reuses job control blocks of completed jobs if it runs out of space for new jobs.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Check the job ID you used and enter the command again if it was incorrect. Enter this command to check the job IDs of all jobs that you have sent to the VM Batch Facility: BATCH STATUS ALL. If you are a VM Batch Facility authorized user, enter this command to check the names of jobs belonging to a particular user: BATCH STATUS *userid nodeid* ALL.

Otherwise, check with the job owner to see if the job has been submitted and completed.

DGRCMD005E Task machine *userid* not found

Explanation: You entered a command specifying a task machine user ID. However, the VM Batch Facility could not find the task machine. The problem has occurred for one of these reasons:

1. The user ID of the task machine was misspelled or incorrect.

2. The task virtual machine has not been identified to the VM Batch Facility. Task machines must be identified with TASK keyword entries in the CONTROL FILE.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Enter the command BATCH QUERY TASK ALL to review the task machines defined to the VM Batch Facility. If the task machine is listed, enter your command again. If not, check with the VM Batch Facility administrator to see if the machine should be a task virtual machine.

DGRCMD006I You are now alert message receiver

Explanation: A VM Batch Facility authorized user has entered the BATCH ALERT command, directing alert messages from the VM Batch Facility to your user ID. You will receive alert messages from now on until:

1. The recipient of alert messages is changed again, or

2. The VM Batch Facility is stopped.

Alert messages report unusual conditions in the VM Batch Facility. You are the only user who will receive all alert messages. If you log off, you should direct alert messages to another user with the BATCH ALERT command. (Note: you must be a VM Batch Facility authorized user to use the BATCH ALERT command.)

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for further commands.

User Response: None is required.

DGRCMD007I VM Batch Facility alert messages are going to *userid*

Explanation: Response to the BATCH ALERT command you entered. The user ID in the message is the user ID currently designated to receive alert messages.

If you entered the BATCH ALERT *userid* command, the message shows that you have successfully directed alert messages to the user ID in the message.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for further commands.

User Response: None is required.

DGRCMD008E Alert messages cannot go to *userid*, **WNG rc =** *rc*

Explanation: You entered the BATCH ALERT *userid* command to send alert messages to a particular user. However, the user you specified cannot receive alert messages. The problem is probably one of these:

1. The user ID is not receiving warning messages. The user may not be logged on, or may be logged on but not receiving warning messages.

2. The user ID was mistyped or not defined in the system.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Reenter the command if the user ID was incorrect. Otherwise, direct alert messages to a user who currently is logged on and receiving warning messages. For more information on the error that occurred, consult the description of the WNG command in your VM library.

DGRCMD009I Console messages are being spooled to *userid* **at** *nodeid*

Explanation: Acknowledges the BATCH CONSOLE command you entered. The user ID in the message is the user ID to which the VM Batch Facility is currently spooling console messages.

If you entered the BATCH CONSOLE *userid nodeid* command, the message means that you have successfully directed console message spooling to the user ID in the message. This user ID will receive the console message log file when the BATCH CONSOLE CLOSE command is entered.

Note: Console message spooling must have been started in order for the program to record console messages. Console message spooling can be started with the CONSOLE keyword in the CONTROL FILE or with the BATCH CONSOLE START command. When console message spooling is started, console messages will be spooled to the user ID named in the message.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD010I Console message spooling started

Explanation: Acknowledges the BATCH CONSOLE START command you entered. The VM Batch Facility has started recording console messages in a spool file. (Console messages record transactions by the VM Batch Facility monitor virtual machine.)

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for other commands.

User Response: None is required.

DGRCMD011I Console message log file closed

Explanation: Acknowledges the BATCH CONSOLE CLOSE command you entered. The VM Batch Facility has closed the console message spool file, if one was found. (A console message log file will be created only if console message spooling has previously been started.)

The file is being punched to the reader of the user ID designated to receive console messages.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for other commands.

User Response: None is required.

DGRCMD012E Console messages cannot go to *userid* **at node id** *nodeid*, **CP SPOOL rc = *rc***

Explanation: You entered the BATCH CONSOLE *userid nodeid* command to change the user ID to which the VM Batch Facility console message log file is being spooled. If you specified the local node, it is likely that the program was unable to spool console messages to the user ID because user ID is not defined on the system. If you specified a remote node, it is likely that the RSCS virtual machine is not correctly specified in your SYSTEM NETID file. The CP SPOOL CONSOLE command failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility was not able to designate the user ID you specified as the console message receiver. Console messages are still being spooled to the previously designated receiver. The program is ready for another command.

User Response: Look up the return code in the message in the description of the CP SPOOL command. CP SPOOL is documented in your VM system library. Correct the problem indicated by the return code and reenter the BATCH CONSOLE command.

DGRCMD013E Console spooling cannot be started, CP SPOOL rc = *rc*

Explanation: You entered the BATCH CONSOLE START command to have the VM Batch Facility start recording console messages. Monitor machine console messages are recorded in the console message log file, a spool file.

However, the program was not able to start console message spooling. The CP SPOOL CONSOLE command failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility was not able to start spooling console messages. Console messages are not being recorded. The program is ready for another command.

User Response: Look up the return code in the message in the description of the CP SPOOL command. CP SPOOL is documented in your VM system library. Correct the problem indicated by the return code and reenter the BATCH CONSOLE START command. The program should then be able to start spooling console messages.

DGRCMD014E Console log file cannot be closed, CP SPOOL rc = *rc*

Explanation: You entered the BATCH CONSOLE CLOSE command to close the VM Batch Facility console message log file. However, the program was unable to carry out the command. The CP SPOOL CONSOLE command failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility was not able to close the console message log file. It could not punch the log file to the user ID designated to receive console messages. Console messages are still being spooled to the designated message receiver, if console message spooling is active.

User Response: Look up the return code in the message in the description of the CP SPOOL command. The SPOOL command is described in your VM system library. Correct the problem indicated by the return code and reenter the BATCH CONSOLE CLOSE command. The program should then be able to close the console message log file.

DGRCMD015I The current CONTROL FILE entries are:

Explanation: Acknowledges the BATCH CONTROL or BATCH CONTROL SET command you entered. The message will be followed by a list showing you the entries in the VM Batch Facility CONTROL FILE.

If you have entered a BATCH CONTROL SET command, the messages show the entries in the CONTROL FILE after your command has been executed. The messages do not include the TASK and CLASS entries in the CONTROL FILE. Use the BATCH QUERY command to review those entries.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD016I Task: *userid* **Classes:** *classid(s)* **Window:** *hhmm–hhmm*
            Status: *current status pending status*
            Active job: *userid nodeid jobid*

Explanation: Acknowledges the following:

- BATCH DRAIN
- BATCH FORCE
- BATCH QUERY TASK
- BATCH RESUME
- BATCH SUSPEND
- BATCH START

Each message shows the status of one task machine. The table below explains the meaning of each part of the message:

This part of the message:	tells you this:
Task: *userid*	The user ID of the task machine described in the message.
Classes: *classid(s)*	The job class(es) the task machine can accept. The task machine can only run jobs that belong to one of the job classes listed in this part of the message.
	If the job class list starts with a period, then each letter or number in the list is a separate type 1 job class classid.
	If the list does not start with a period, then Classes shows the classid(s) of type 2 job classes that are accepted by the task machine. For example, if you see "Classes: large" in the message, that means that the task machine can run jobs that have been assigned to any of these job classes: LA, LAR, LARG, and LARGE. Task machines can accept jobs from any class whose class ID is included in the Classes: value, reading from the left of the value.
Window: *hhmm–hhmm*	The start time window of the task machine. The task machine is available for starting jobs only between the times shown after Window. If the times are 0000 and 2400, then the task machine is available for starting jobs 24 hours a day.

This part of the message:	tells you this:
Status: *current status pending status*	The current status and pending status of the task machine.
	The current status is the condition of the task machine right now. The pending status is the condition that is about to take over in the task machine. For example, after a BATCH DRAIN command, the status may be RUNNING HOLD. That means that the task machine is running a job right now, but will be put into HOLD status as soon as the job is over.
	The current and pending status may be:
	• **Deleted**–the task machine's TASK entry has been deleted from the CONTROL FILE. Jobs can no longer be started on the task machine.
	• **Hold**–the task machine is unavailable for running jobs as a result of a BATCH DRAIN or BATCH FORCE HOLD command from an authorized user. The task machine must be made available with the BATCH START command before it can be used to run jobs again.
	• **Offline**–the task machine is unusable. The VM Batch Facility was unable to AUTOLOG the task machine to run a job.
	• **Purging job**–the task machine is being AUTOLOGged to do ERROR option processing for a canceled job.
	• **Ready**–the task machine is ready to run a job.
	• **Running**–the task machine is running a job.
	• **Suspended**–the job that was running has been temporarily halted with the BATCH SUSPEND command.
	• **Task logging off**–the task machine has finished a job and is logging off.
	• **To be restarted**–the task machine is being restarted by the VM Batch Facility to run a job that was interrupted by a system shutdown.
	• **Wait for JOBEND**–the task machine has logged off after its job was canceled and is about to be AUTOLOGged again to perform ERROR option processing. The VM Batch Facility is waiting for all processing to be completed.
	• **Window not open**–the task machine cannot start jobs right now. The current time does not fall within the start time window defined for the task machine in its TASK keyword entry.
Active job: *userid nodeid jobid*	Which job is currently being run by the task machine. The line will appear only when a job is running.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

Note that the format of this message is slightly different when issued by a monitor on a remote node. However, the content is the same.

DGRCMD017I *END OF STATUS

Explanation: You entered one of the forms of the BATCH STATUS command and have received status messages on one or more jobs. This message marks the end of the job status messages sent in response to your command. The message may be useful to you if you use a programmable operator or other application and would like to trap job status messages.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD018E No jobid entered

Explanation: You entered the BATCH STATUS JOB command, but there was no job ID following JOB.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: To get information on a specific job, reenter the command, specifying the job ID of the job on which you want information. Use the JOB option of the BATCH STATUS command only if the job ID is the same as one of the BATCH STATUS command options or abbreviations.

DGRCMD019E No password entered

Explanation: You entered the BATCH PASSWORD command but did not supply your logon password in the command.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Enter the command again, specifying your logon password. The command will associate the new password with all of the CLASS AUTOLOG jobs that you have submitted from this user ID.

DGRCMD020E Password rejected - CLASS AUTOLOG job not found

Explanation: You entered the BATCH PASSWORD command, but you do not have any CLASS AUTOLOG jobs in the VM Batch Facility. The problem may be that:

1. Your CLASS AUTOLOG jobs have been completed or canceled.

2. You entered the command from a user ID other than the one you were on when you submitted a CLASS AUTOLOG job or jobs. The program keeps track of jobs by user ID.

3. You have not submitted any CLASS AUTOLOG jobs.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Check to be sure that you entered the command from the user ID you were on when you submitted your CLASS AUTOLOG jobs. Enter the command again from the correct user ID if necessary.

Otherwise, no action is required since your CLASS AUTOLOG jobs have been completed or canceled.

DGRCMD021I Password received

Explanation: Acknowledges the BATCH PASSWORD command you entered. The VM Batch Facility has stored the new password you supplied in the command for use in running all CLASS AUTOLOG jobs that you submitted from this user ID.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD022I BATCH PRIORITY command successfully completed

Explanation: Acknowledges the BATCH PRIORITY command you entered. The VM Batch Facility has reset the priority of the job you identified in the command.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD023E Job *userid nodeid jobid* **is not queued to run**

Explanation: You entered the BATCH PRIORITY command to change the priority of a job. The VM Batch Facility could not change the job's priority because the job is not queued to run. The most likely cause of the problem is that the job has already begun to run or has been completed or canceled.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: To check on the job you specified, use the BATCH STATUS command. Otherwise, no response is required.

DGRCMD024I No VM Batch Facility task machines defined

Explanation: You entered a BATCH QUERY TASK ALL command to review VM Batch Facility task machines. However, no task virtual machines have been defined to the VM Batch Facility. Task machines are virtual machines used by the program to run jobs.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: Contact the VM Batch Facility administrator to find out whether task machines should be defined.

DGRCMD025I No VM Batch Facility task machines are running

Explanation: You entered the BATCH QUERY TASK RUNNING command to review running task machines. No task machines are running jobs at this time.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD026I No VM Batch Facility task machines are ready

Explanation: You entered the BATCH QUERY TASK READY command to review task machines that are waiting for jobs but do not have jobs to run or that are waiting for their start time windows to open. No task machines are in this condition at this time.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD027I No VM Batch Facility task machines are in hold

Explanation: You entered the BATCH QUERY TASK HOLD command to review task machines that are in HOLD status because of a BATCH DRAIN or BATCH FORCE HOLD command. No task machines are in HOLD status at this time.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD028I Class: *classid* **Window:** *hhmm–hhmm*
           Virtual Storage: *dddddddK/nnnnnnnK/mmmmmmmK*
           Seconds: *ssssssss/ssssssss* **Print:** *ppppK/ppppK* **Punch:** *ppppK/ppppK*
           Minimum Stall Delay: *cmsd* **Maximum Stall Warnings:** *cmsw*
           LLS subject: *llss*   **LLS overrides:** *llso*
           Jobs queued = *nnnn*, **Jobs running =** *nnnn*

Explanation: The response to a BATCH QUERY CLASS command. Each message shows the definition of one job class. The table below explains the meaning of each part of the message:

This part of the message:	tells you this:
Class: *classid*	This shows you the class ID (or name) of the job class described in the message.
Window: *hhmm–hhmm*	This is the start time window defined for the job class.
	The job class start window provides default start window times for jobs assigned to the class. If a job owner does not use the Start Window Open or Start Window Close fields (WBEGIN or WEND options), then the appropriate job class time(s) apply to the job.
Virtual Storage: *dddddddK/nnnnnnnK/ mmmmmmmK*	This shows the amount of virtual storage available by default (*dddddddK*), and the minimum (*nnnnnnnK*) and maximum (*mmmmmmmK*) values that can be assigned to a job of this class. If the size is specified in Megabytes, the terminating character shown is 'M' rather than 'K'.
	If the default value is shown as an asterisk (*), this means that jobs of this class run with the CP directory value of virtual storage in the allocated task machine.
	If the amount shown for minimum and maximum storage is an asterisk (*) this means that jobs of this class are not permitted to request storage amount other than the class default.
Seconds: *ssssssss/ssssssss*	This shows the default limit and maximum limit on the number of CPU seconds that a job can use. The default limit is the first number; the maximum is the second.
	If a job is submitted without a value in the CPU Seconds field (SECONDS option), the job will be subject to the default job class limit.
	If a job is submitted with a CPU Seconds or SECONDS value that is higher than the class maximum, the value will be reset to the class maximum.
	-/NONE is reported for jobs with no limit.
	If a job exceeds the limit that applies to it, it may be canceled.
Print: *ppppK/ppppK*	This shows the default limit and maximum limit on the records that a job can send to the virtual printer of a task machine. The default limit is the first number; the maximum is the second.
	If a job is submitted without a value in the Print Records field (PRINT option), the job will be subject to the default job class limit.
	If a job is submitted with a Print Records or PRINT value that is higher than the class maximum, the value will be reset to the class maximum.
	-/NONE is reported for jobs with no limit.
	If a job exceeds the limit that applies to it, it may be canceled.

This part of the message:	tells you this:
Punch: *pppp*K/*pppp*K	This shows the default limit and maximum limit on the records that a job can send to the virtual punch of a task machine. The default limit is the first number; the maximum is the second.
	If a job is submitted without a value in the Punch Records field (PUNCH option), the job will be subject to the default job class limit.
	If a job is submitted with a Punch Records or PUNCH value that is higher than the class maximum, the value will be reset to the class maximum.
	-/NONE is reported for jobs with no limit.
	If a job exceeds the limit that applies to it, it may be canceled.
Minimum Stall Delay: *cmsd*, Maximum Stall Warnings: *cmsw*	This shows the values of MINIMUM-STALL-DELAY and MAXIMUM-STALL-WARNINGS that are in effect for jobs in the class.
LLS subject: *llss*, LLS overrides: *llso*	This shows the default LLS subjection of jobs in the class, and whether the default subjection may be overriden.
Jobs queued = *nnnn*, Jobs running = *nnnn*	This line tells you how many jobs are queued and waiting to run in the job class, and how many jobs of the class are being run.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

Note that the format of this message is slightly different when issued by a monitor on a remote node. However, the content is the same.

DGRCMD029E Job class *classid* not defined

Explanation: You entered the BATCH QUERY CLASS *classid* command. However, the class that you identified has not been defined to the VM Batch Facility.

The problem may have occurred for either of these reasons:

1. The class ID was misspelled or incorrect in the command.

2. The class has not been defined to the VM Batch Facility. Job classes must be defined in the CONTROL FILE. And, if changes have been made to it, the CONTROL FILE must have been put into effect by starting the program or issuing the BATCH CONTROL SET command.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for other commands.

User Response: Correct the command and reenter it if the class ID was in error. Otherwise, enter this command to review all job classes defined to the VM Batch Facility: BATCH QUERY CLASS ALL. If the class you specified is not listed, it is not defined to the program and is not an existing job class.

DGRCMD030I BATCH START command processed

Explanation: Acknowledges the BATCH START command that you entered. The message indicates that a BATCH STOP command has been successfully canceled, if one was pending.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD031I The monitor will stop when all running jobs have ended

Explanation: Acknowledges the BATCH STOP command that you entered. VM Batch Facility task machines will log off as each machine finishes the job it is currently running. The VM Batch Facility control program, running in the monitor machine, will stop when the last task machine logs off.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD032I The monitor is stopping now; all running jobs continue

Explanation: Acknowledges the BATCH STOP NOW command that you issued. The VM Batch Facility control program, in the monitor machine, will stop running right away. The monitor machine remains logged on.

VM Batch Facility task machines will log off as each machine finishes the job it is currently running.

VM Batch Facility Action: Command completed. As soon as the control program stops running, the VM Batch Facility will not respond to any commands.

User Response: None is required.

DGRCMD033I The monitor is stopping now; all running jobs are being canceled

Explanation: Acknowledges the BATCH STOP JOBS command that you issued. The VM Batch Facility control program, in the monitor machine, will stop right away. All running task machines will also log off right away. The running jobs will be canceled and will not be restarted when you start the VM Batch Facility again.

VM Batch Facility Action: The VM Batch Facility has completed the command. As soon as the control program stops running, the program will not respond to any commands.

User Response: None is required.

DGRCMD034I *userid nodeid jobid*
       Status: *mm/dd/ccyy hh:mm:ss status*
       Class: *classid* **Schedule:** *mm/dd/ccyy hhmm-hhmm* **Step:** *jobstep*
       Seconds: *sssssss/sssssssss*, **Print:** *ppppK/ppppK*, **Punch:** *ppppK/ppppK*
       Virtual Storage: *sssssssK*
       Queue position is *n* **within job class** *classid*
       Chained to: *userid nodeid jobid jobstep*
       Job running in task machine *taskid*
       Started: *mm/dd/ccyy hh:mm:ss*
       Recurrence pattern: *ffffffff* **[(HOLD)]**
       Recurrence is set by chain. [(HOLD)]
       Occurrence count: *nnnnn*
       Duplicate jobs have been rejected

Explanation: The response to a BATCH STATUS command. It may also appear in some other circumstances, as when you submit a job that has the same job ID as a completed job. The VM Batch Facility will display the status of the completed job before erasing its records to make way for the new job.

The message only contains those lines that are appropriate for the job. For example, only the job identification and the reportable status are displayed for jobs that have been rejected.

If a recurring job is running, or ready to run, or has been held, then the message reports the execution or queue status of the current instance. Otherwise, the message reports the completion status of the last executed instance (if any).

Each message shows the status of one job. The table below explains the meaning of each part of the message.

This part of the message:	tells you this:
userid	The user ID from which the job was submitted.
nodeid	The node ID from which the job was submitted.
jobid	The job ID of the job–the name that has been given to the job.
	The job ID may be the time of day when the job was submitted, as in 12023522. (The time is shown in hours, minutes, seconds format.)
	Or it may be a name assigned to the job by the job owner.
	To direct commands specifically to a job, the command must include the job ID of the job. And the VM Batch Facility refers to a job by its job ID in any messages it sends about the job.
Status: *mm/dd/ccyy hh:mm:ss status*	The current status of the job. The date (*mm/dd/ccyy*) and time (*hh:mm:ss*) show when the job reached its current status. For RUNNING jobs, the date and time shown indicate when the jobs were last seen to be running. (Note that the date may also appear in dd.mm.ccyy format.)
	The words after the date and time indicate the current condition of the job. The status may be:
	• **Admin-held**–the job has been withheld from execution by command from a VM Batch Facility administrator.
	• **Autologged**–this CLASS AUTOLOG job has been started. The job is no longer under the control of the VM Batch Facility.
	• **Canceled**–the job has been discarded and will not run. The status will also tell you why or by whom the job was canceled.
	• **Completed**–the job has been completed.
	• **Exit-held**–the job has been withheld from execution by command from a VM Batch Facility exit.
	• **Hold**–the job cannot run because there is no task machine configured to accept it.
	• **Forced by** *userid*–the job has been canceled by a BATCH FORCE command from a VM Batch Facility authorized user.
	• **Logged or forced off**–the job has been canceled by the VM system or by a LOGOFF command in the job exec.
	• **Queued**–the job is waiting to run.
	• **Restarted**–the job has been restarted. The job had been interrupted by a system shutdown while it was running.
	• **Running**–the job is currently being performed.
	• **Skipped**–the job was not run at the time scheduled because it was held due to a **REPEAT HOLD** or a **DATE HOLD**.
	• **Stalled mm:ss**–the job has been stalled for the indicated duration.
	• **Started**–the task machine has been logged on to run a job.
	• **Suspended**–the job has been temporarily halted by a VM Batch Facility authorized user, or by the Load Level Scheduler.
	• **Rejected**–the reason for rejection is included in the message. Only the first two message lines are shown for rejected jobs.
	• **Awaiting synchronization**–the WAIT option was specified and synchronization of the submission has not yet completed.
	• **Flushed**–the recurring job has been flushed and will not recur. The status will also tell you by whom the job was flushed.
	• **Synchronization timed out**–a synchronous submission was abandoned or timed out.
	• **Prevented**–an occurrence of a recurring job has been prevented from running by the DGRECUR exit not confirming its start.
Class: *classid*	The job class to which the job has been assigned.

This part of the message:	tells you this:
Schedule: *mm/dd/ccyy hhmm-hhmm*	The start date and start time window defined for the job.
	The job is eligible to start on the date shown in the message. (Note that the date may also appear in dd.mm.ccyy format.) It can be started only between the first and second times shown after the start date. (These two times define the start time window of the job. The first time is the Start Window Open or WBEGIN time. The second time is the Start Window Close or WEND time.)
	If HOLD appears instead of a date, the job is to be held until the job owner specifies a start date for it.
	If OPEN appears instead of the start window close time, the job can be started anytime after the start window open time.
Step: *jobstep*	The text of the last BATCH JOBSTEP command that was executed by the job. There will not be any *jobstep* if the job has not started or has not executed any BATCH JOBSTEP commands.
Seconds: *sssssss/sssssss*	The first *sssssss* shows how many seconds of CPU time the job has used.
	The second *sssssss* shows the limit on CPU seconds that applies to this job. The job may be canceled if the number of seconds it has used exceeds this limit.
	-/NONE is reported for jobs with no limit.
Print: *ppppK/ppppK*	The first *pppp* shows many thousands of records the job has sent to the virtual printer.
	The second *pppp* shows the limit on print records that applies to this job. The job may be canceled if the number of records that it has produced, in thousands, exceeds this limit.
	-/NONE is reported for jobs with no limit.
Punch: *ppppK/ppppK*	The first *pppp* shows how many thousands of records the job has sent to the virtual punch.
	The second *pppp* shows the limit on punch records that applies to this job. The job may be canceled if the number of records that it has produced, in thousands, exceeds this limit.
	-/NONE is reported for jobs with no limit.
Virtual Storage: *sssssss*K	This shows the amount of virtual storage allocated to your job, either by default or as you specified with the STORAGE option on the SUBMIT or CHJOB command.
	If the value is in megabytes the last character will be 'M' instead of 'K'. If shown as an asterisk (*), the TASK machine default has been used.
Queue position is *n* within job class *classid*	This line appears when a job is ready to run, and shows how many jobs are in line in front of this job. A job is the next job to run when the queue position is 1.
	The line will appear only when the start date of the job has arrived and its start time window has opened.
Chained to: *userid nodeid jobid jobstep*	This line appears only for jobs that have been chained to other jobs. A chained job cannot start until the job to which it is chained has reached a specified point in its execution. The user ID, node ID, job ID and jobstep in the message identify the job to which this job is chained. The line will not appear if a chained job has already started to run.
Job running in task machine *userid*	This line shows you which task machine is running the job. The line will appear in the message only when a job is running.
Started: *mm/dd/ccyy hh:mm:ss*	This shows when the job started running.
Recurrence pattern: *ffffffff* [(HOLD)]	This shows the recurrence pattern defined for the job if the job recurs explicitly. The presence of (HOLD) indicates that recurrence of the job has been suspended.

This part of the message:	tells you this:
Recurrence pattern is set by chain. [(HOLD)]	This indicates a job that recurs implicitly. The presence of (HOLD) indicates that recurrence of the job has been suspended.
Occurrence count: *nnnnn*	This shows the number of times the job has started.
Duplicate jobs have been rejected	This indicates that at least one job with the same ID was submitted by the same user while the job was active, and was rejected for that reason.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for another command.

User Response: None is required.

Note that the format of this message is slightly different when issued by a monitor on a remote node. However, the content is the same.

DGRCMD036E Task *userid* **cannot be suspended, CP SMSG rc =** *rc*

Explanation: You entered the BATCH SUSPEND command specifying the task machine named in the message. However, the VM Batch Facility could not suspend the task machine. The CP SMSG command issued by the monitor machine failed with the return code shown in the message.

VM Batch Facility Action: Command not performed. The task machine has not been suspended. The VM Batch Facility is ready for other commands.

User Response: Look up the return code in the message in the description of the CP SMSG command. The SMSG command is described in your VM system library. When you have corrected the problem, reenter the BATCH SUSPEND command.

The problem may be that the SET SMSG OFF command has been issued by the job that is running in the task machine. Or, the task machine named in the message may be in HOLD status. Use the BATCH QUERY TASK HOLD command to see if the task machine is in HOLD.

DGRCMD037E Task *userid* **cannot be resumed, CP SMSG rc =** *rc*

Explanation: You entered the BATCH RESUME command to restart a suspended task machine. However, the VM Batch Facility could not restart the task machine. The CP SMSG command issued by the monitor machine failed with the return code shown in the message.

VM Batch Facility Action: Command not performed. The suspended task machine has not been restarted. The VM Batch Facility is ready for other commands.

User Response: Look up the return code in the message in the description of the CP SMSG command. The SMSG command is described in your VM system library. When you have corrected the problem, reenter the BATCH RESUME command.

DGRCMD038I Console message spooling stopped

Explanation: Acknowledges the BATCH CONSOLE STOP command that you entered. The VM Batch Facility has stopped recording messages in the console message log file.

VM Batch Facility Action: Command completed. The VM Batch Facility is ready for other commands.

User Response: None is required.

DGRCMD039E Console spooling cannot be stopped, CP SPOOL rc = *rc*

Explanation: You entered a BATCH CONSOLE STOP command to stop the VM Batch Facility from recording messages in the console message log file. However, the program was not able to carry out your command. The SPOOL CONSOLE STOP command that the program issued failed with the return code shown in the message.

VM Batch Facility Action: Command not performed. Console messages are still being recorded in a spool file for the designated message receiver. The VM Batch Facility is ready for other commands.

User Response: Look up the return code in the message in the description of the CP SPOOL command. The command is described in your VM system library. Correct the problem indicated by the return code and reenter the BATCH CONSOLE STOP command.

DGRCMD040W Invalid return code from DGRUJB user exit, rc = *rc*

Explanation: The VM Batch Facility has received an invalid return code from the job screening exit, DGRUJB EXEC. Only the return codes documented in the prolog of DGRUJB EXEC and in the *VM Batch Facility Installation, Customization, and Administration* manual can be accepted by the VM Batch Facility.

VM Batch Facility Action: The VM Batch Facility has ignored the invalid return code. The job that was being processed will not be affected by the return code. It will be processed in normal fashion.

User Response: Ensure that the DGRUJB EXEC uses only the documented return codes.

DGRCMD041E *command* **cannot be set to a task machine**

Explanation: One of these problems has occurred:

* While starting or after a BATCH CONTROL SET command, the VM Batch Facility found that the CONTROL FILE designates a task machine as the alert or console message receiver.

* Or, during operation, the BATCH ALERT *userid* or BATCH CONSOLE *userid* command designated a task machine as the alert or console message receiver.

Alert and console messages cannot be directed to a task machine.

VM Batch Facility Action: If the problem occurred when the VM Batch Facility was starting or after a BATCH CONTROL SET command was entered:

1. The VM Batch Facility has designated the ID of the monitor machine as the console message receiver and is continuing to operate, if the problem was with the CONSOLE entry.

2. If the problem was with the alert message receiver and was found while the VM Batch Facility was starting, the VM Batch Facility has shut down.

   If the problem was found after a BATCH CONTROL SET command, the VM Batch Facility has designated the person who entered the command as the alert message receiver and as an authorized user. The program is continuing to operate.

If the problem occurred when a BATCH ALERT or BATCH CONSOLE command was entered, the command has not been carried out. The VM Batch Facility is ready for another command.

User Response: If the problem occurred while starting or after a BATCH CONTROL SET command was entered, correct the ADMIN or CONSOLE entry in the CONTROL FILE. The first ADMIN entry in the CONTROL FILE designates the alert message receiver. The CONSOLE keyword entry designates the console message log file receiver.

Then enter the BATCH CONTROL SET command or restart the VM Batch Facility to put the CONTROL FILE changes into effect.

If the problem occurred when you entered the BATCH ALERT or BATCH CONSOLE command, enter the command again, specifying a user ID that does not belong to one of the VM Batch Facility task machines.

DGRCMD045I through DGRCMD065I

Explanation: DGRCMD messages from 045I to 065I record commands issued to the VM Batch Facility. You will not see these messages unless you review the console log file, where the messages are recorded. Each message records a command that the program has received and the user ID from which the command was entered. The following table lists the command on which each message reports:

This message:	Records the issuance of this command:
DGRCMD045I	Any authorized command issued by a user who is not an authorized user. Commands are not executed.
DGRCMD046I	Any invalid command--a command not recognized by the VM Batch Facility or by the user command screening exit, DGRUCD EXEC.
DGRCMD047I	BATCH ALERT command.
DGRCMD048I	BATCH CONSOLE command.
DGRCMD049I	BATCH CONTROL command.
DGRCMD050I	BATCH DRAIN command.
DGRCMD051I	BATCH FORCE command.
DGRCMD053I	BATCH PASSWORD command.
DGRCMD054I	BATCH PRIORITY command.
DGRCMD055I	BATCH QUERY command.
DGRCMD060I	BATCH STOP command.
DGRCMD061I	BATCH STATUS command.
DGRCMD062I	BATCH SUSPEND command.
DGRCMD064I	BATCH RESUME command.

VM Batch Facility Action: These messages are recorded in the console message log file, a spool file, only if console logging is in effect.

User Response: None is required. To view these messages, enter the BATCH CONSOLE CLOSE command. The console log file will be closed and sent to the designated console message receiver. A new console log file will automatically be started if console message spooling is active.

DGRCMD066I No jobs found

Explanation: Acknowledges a BATCH STATUS command and indicates that:

- The VM Batch Facility does not have any jobs belonging to you, if you entered the BATCH STATUS ALL command.
- Or, there are no jobs in the VM Batch Facility, if you entered another form of the BATCH STATUS command.

VM Batch Facility Action: The command has been performed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD067I No *option* **jobs found**

Explanation: Acknowledges a BATCH STATUS command. The VM Batch Facility has searched its status records, but did not find any jobs matching the BATCH STATUS option you specified in the command. For example, if you enter BATCH STATUS QUEUED and receive this message, then none of your jobs is waiting to run.

VM Batch Facility Action: The command has been performed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD068I The monitor machine is stopping now

Explanation: This message is issued to indicate that the monitor is stopping. It is preceeded by a message indicating why it is stopping.

VM Batch Facility Action: The VM Batch Facility is shutting down.

User Response: None.

VM Batch Facility Action: The VM Batch Facility will shut down.

User Response: None is required.

DGRCMD069E Invalid priority specified

Explanation: The BATCH PRIORITY command that you entered specified an invalid priority. The priority must be a number from 0 to 9.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Enter the command again, specifying a priority number in the range from 0 to 9. 0 is the highest priority. 9 is the lowest priority and is also the default priority.

DGRCMD070I *nnn* **jobs cleared**

Explanation: This message appears as a response to a BATCH STATUS command with the CLEAR option.

VM Batch Facility Action: The command has been performed. The VM Batch Facility is ready for another command.

User Response: None is required.

DGRCMD5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGREXE101E You used too many options in your command; *option* **is extraneous**

Explanation: You used too many options in the VM Batch Facility command that you entered. The message shows which option could not be used.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE103E Value *value* **given for option** *option* **must be** *type*

Explanation: One of the values you specified in the command you entered was invalid because it was the wrong type of value. The problem may have been, for example, that you entered letters for an option that requires a numeric value. The message indicates what type of value must be entered for the option.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE104E Required value *value* **missing for** *command* **command**

Explanation: One of these problems was found in the command that you just entered:

- A required option was missing from the command.
- No value was specified for one of the options that you used and a value is required.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE105E You must specify a *jobid* **before optional parameters**

Explanation: There was an error in the format of the BATCH CHJOB command that you just entered. The command format must be:

```
batch chjob jobid (options
```

with the job ID of the job to be changed coming before any options.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE107E There were invalid option(s) in the *command* **command**

Explanation: The VM Batch Facility did not recognize one or more of the options that you used in the command named in the message.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE108E [command/option**] [**option/value**] is missing or invalid**

Explanation: There was an error in the BATCH SUBMIT or BATCH CHJOB command that you entered. The message shows which command, option, and/or value was incorrect or missing.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE109E *value* **must be no greater than** *n* **characters long**

Explanation: A value in the command you just entered was too long. A user ID, for example, can only be up to eight characters long.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format or enter the command name followed by a question mark. Then enter the command again.

DGREXE110E Maximum number of append files is nine

Explanation: The VM Batch Facility is able to work with up to nine files listed in the APPEND option of the BATCH SUBMIT command. The command you entered listed more than nine files for the VM Batch Facility to append.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: List some or all of the files to be appended in an APNDLIST file. You can use an APNDLIST file to list as many files to append as you would like. For more information on using APNDLIST files, see the *VM Batch Facility User's Guide*, or type in BATCH SUBMIT ? for online help.

DGREXE111E The value of *command/option* **must be in the range from** *n1* **to** *n2*

Explanation: The value that you entered for the command or option named in the message was too large or too small.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Enter the command again, using a value that is within the range shown in the message. For more help with the command, type in the command name followed by a question mark or see the *VM Batch Facility User's Guide.*

DGREXE112E [*option/value* **] specified with [** *command/option* **] is invalid**

Explanation: One of these problems has been found in the BATCH SUBMIT, BATCH CHJOB, or BATCH PASSWORD command you entered:

1. The VM Batch Facility did not recognize one of the SUBMIT or CHJOB options or values that you entered in the command.
2. You used the REPEAT option with a CLASS AUTOLOG job.
3. You used the APPEND option with a CLASS AUTOLOG job.
4. You specified NOLOG as the password in a BATCH PASSWORD command or in the PASSWORD option of BATCH SUBMIT or BATCH CHJOB. The VM system does not allow NOLOG as a password.
5. You specified a file name and file type on a BATCH CHJOB command. The BATCH CHJOB command must specify the job ID of a job, not the file name or file type.
6. There was no left parenthesis before the options in a BATCH CHJOB command. Job control options must be preceded by a left parenthesis, as in: "batch chjob myjob (print 50 punch 50 ...." If this problem has occurred, the VM Batch Facility will interpret the first option in the command as a file type, and the message will indicate that the file type that you specified is not valid.

The option or value that must be changed is shown in the message.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the command format in VM Batch Facility *User's Guide* or enter the command name followed by a question mark. Correct the command and enter it again.

DGREXE113E *filename* **is not present on any of your accessed disks**

Explanation: The VM Batch Facility could not find the file named in the message on any of the minidisks available to your user ID.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: If the file name in the message is DGRUVM, contact the VM Batch Facility administrator for help in getting access to the minidisk on which the file is stored.

Otherwise, ensure that you have access to the minidisk on which the file is located and enter the command again.

DGREXE115E [Filename/Filetype/Filemode] of appended file *fn ft fm* **is invalid, must be no more than** *n* **characters**

Explanation: The file name, file type, or file mode of the file named in the message is too long. The file is a file that you listed with the APPEND option of the BATCH SUBMIT command or Submit panel.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Enter the command again. Be sure that the file name is entered correctly.

DGREXE118E A job exec filename is required for class other than AUTOLOG

Explanation: You entered a BATCH SUBMIT command but did not include the file name of the job exec that is to be submitted. A job exec name must be specified in the command unless the job is being submitted to the AUTOLOG job class.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Enter the command again. Include the file name of the job exec that is to be run or specify AUTOLOG as the value of the CLASS option.

DGREXE119I Job *jobid* **submitted to** *monitor* **at** *node*

Explanation: Acknowledges that you have submitted a job to the VM Batch Facility. The message shows the job ID, or job name, that has been assigned to the job; and the name of the monitor and node to which the job was submitted.

Make a note of the job ID for future reference. You will need to use the job ID if you wish to address VM Batch Facility commands to this job alone. Also, the VM Batch Facility will reference the job ID in any messages it sends you about the job.

VM Batch Facility Action: The VM Batch Facility has carried out your BATCH SUBMIT command. It is ready for another command.

User Response: None is required.

DGREXE120I You are now addressing Monitor *monitor* **at node** *node*

Explanation: This message is issued in response to a BATCH ADDRESS command. It identifies the VM Batch monitor (and its location) that your VM Batch jobs and commands will be submitted to.

VM Batch Facility Action: Until the BATCH ADDRESS command is entered again, all VM BATCH jobs and commands will be directed to the monitor identified in the message

User Response: None is required.

DGREXE121E The SYSTEM NETID is not available for node identification

Explanation: Your jobs and commands have been directed to a monitor at a remote node. The identification of the RSCS virtual machine on your own node must be available in order for communication to take place; but this identification has not been found on your system, or is invalid.

VM Batch Facility Action: No commands are passed to remote monitors when no local RSCS virtual machine identification is available.

User Response: Correct or create the SYSTEM NETID file.

DGREXE122E The RSCS virtual machine *rscsid* **is not logged on**

Explanation: You are addressing your jobs and commands to a VM Batch Facility on a remote node, but the RSCS virtual machine is not logged on.

VM Batch Facility Action: No remote VM Batch Facilities are available to you until the RSCS virtual machine is logged on.

User Response: Confirm that your addressing of a remote VM Batch Facility is correct. If not, change your destination VM Batch Facility using the BATCH ADDRESS command. Otherwise contact your system support personnel to ensure that the RSCS virtual machine is correctly identified in the SYSTEM NETID file, and that the virtual machine is logged on.

DGREXE123E The VM Batch monitor *monitorid* **is not available**

Explanation: You are addressing your jobs and commands to a VM Batch Facility monitor on this node that is not logged on.

VM Batch Facility Action: This VM Batch Facility is not available to you until the monitor virtual machine is logged on.

User Response: Confirm that your addressing of this VM Batch Facility monitor is correct. If not, change your destination VM Batch Facility using the BATCH ADDRESS command. Otherwise contact your system support personnel to ensure that the VM Batch Facility monitor virtual machine is correctly identified in the VMBATCH DEFAULTS file, and that the virtual machine is logged on.

DGREXE124E The password could not be read due to a *macro* **error**

Explanation: While prompting for the user's password during the submission of a CLASS AUTOLOG job, DGRIPW MODULE has detected an unrecoverable error executing the CMS *macro* function.

VM Batch Facility Action: The submission of the CLASS AUTOLOG job is canceled.

User Response: Re-IPL CMS in your virtual machine and submit the CLASS AUTOLOG job again. If problems persist, contact your VM Batch Facility Administrator.

DGREXE5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRINI001I The VM Batch Facility is starting

Explanation: Displayed at the monitor machine console and recorded in the console message log file, when the VM Batch Facility is starting.

VM Batch Facility Action: The VM Batch Facility is starting. If start-up is successful, the VM Batch Facility will be ready for use shortly.

User Response: None is required.

DGRINI004I Virtual storage size = *nnnnnnn***K**

Explanation: The virtual storage size of the monitor virtual machine, in which the VM Batch Facility control program runs.

VM Batch Facility Action: Start-up is proceeding normally. The message does not indicate that a problem has occurred.

User Response: None is required.

DGRINI005E Error on FSSTATE command for CONTROL FILE, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to find the CONTROL FILE. The CONTROL FILE may be missing from the monitor machine's 199 minidisk, or the minidisk may not be accessed.

VM Batch Facility Action: The VM Batch Facility has shut down. It is unable to start without the CONTROL FILE.

User Response: Look up the return code in the message in the description of the CMS FSSTATE macro. The FSSTATE macro is described in your VM system library.

To check on likely causes of the error, the VM Batch Facility monitor machine, and check to be sure that the 199 minidisk has been accessed as disk G. Also check to be sure that the CONTROL FILE is located on the 199 minidisk.

DGRINI006E Error on DMSFREE call, rc= *rc*

Explanation: The DMSFREE macro issued by the VM Batch Facility control program failed with the return code shown in the message. The most likely cause of the problem is that the VM Batch Facility monitor machine did not have sufficient unused virtual storage for the VM Batch Facility to operate.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the DMSFREE macro. The macro is described in your VM system library. Correct the problem indicated by the return code, and restart the VM Batch Facility. For instructions on calculating the virtual storage required for the monitor virtual machine, see "Tailoring the Monitor Machine to Handle Higher Work Loads" on page 45.

DGRINI007E Job Class A not found in CONTROL FILE

Explanation: Job class A is not defined in the VM Batch Facility CONTROL FILE. Job class A is the default job class, used if a job owner does not specify another job class for a job.

Job class A must be defined if any TASK keywords are present in the CONTROL FILE, defining task virtual machines. If the VM Batch Facility is set up to accept only CLASS AUTOLOG jobs and no task machines are defined, class A is not required.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: If you wish to run the VM Batch Facility without job class A, edit the CONTROL FILE. Ensure that there are no TASK keywords, and that the only job class is AUTOLOG.

Otherwise, edit the CONTROL FILE and add a CLASS entry to define job class A. Refer to "Adding and Deleting Job Classes" on page 34 for more information.

DGRINI009E No valid administrators defined in CONTROL FILE

Explanation: The VM Batch Facility CONTROL FILE does not define any authorized users, users who can enter the privileged commands required for operating and administering the VM Batch Facility. The VM Batch Facility requires at least one authorized user.

The problem has occurred for one of these reasons:

1. There are no ADMIN keyword entries in the CONTROL FILE. ADMIN entries identify authorized users to the VM Batch Facility.

2. The user IDs identified with ADMIN keywords do not exist on this system, or no user IDs were specified in the ADMIN entries.

VM Batch Facility Action: If the error occurs when the VM Batch Facility is starting, the VM Batch Facility shuts down. It is unable to operate without at least one authorized user.

If the error occurs after a BATCH CONTROL SET command has been entered, the VM Batch Facility continues operation. The VM Batch Facility will temporarily use the user ID of the person who issued the BATCH CONTROL SET command as the only authorized user.

User Response: Edit the CONTROL FILE and add at least one ADMIN keyword and user. Ensure that the user IDs are spelled correctly and are defined in the CP directory of your system. Be sure to make this correction before stopping and restarting the VM Batch Facility. The VM Batch Facility will not be able to start if no valid administrators are defined.

DGRINI010E No valid task machines defined in CONTROL FILE

Explanation: In processing the CONTROL FILE, the VM Batch Facility found that there were no valid TASK keyword entries in the file. TASK keywords identify task virtual machines to the VM Batch Facility.

The problem has occurred for one of these reasons:

1. There were no TASK keyword entries in the CONTROL FILE.

2. One or more values in the TASK entries were invalid or missing.

The VM Batch Facility does not have to have any task machines if the AUTOLOG job class is the only class defined in the CONTROL FILE.

VM Batch Facility Action: If the error occurs when the VM Batch Facility is starting, it shuts down. It is unable to operate without at least one task machine, except when only CLASS AUTOLOG is defined as a job class.

If the error occurs after a BATCH CONTROL SET command has been issued, the VM Batch Facility continues operation. However, it will not have any task machines available to it.

User Response: Edit the CONTROL FILE and ensure that there is at least one valid TASK entry in the file. Then restart the VM Batch Facility or issue the BATCH CONTROL SET command to put the corrections into effect.

DGRINI011E No valid classes defined in CONTROL FILE

Explanation: The VM Batch Facility found one of these problems with the CONTROL FILE:

1. There are no CLASS entries in the file or no valid CLASS entries. CLASS entries define job classes.

2. There is only one job class defined in the CONTROL FILE and it is not either CLASS A or CLASS AUTOLOG.

3. There are multiple job classes defined, but no job class A.

The CONTROL FILE must define at least one job class. If there is only one job class, it must be either A or AUTOLOG. And if there are multiple job classes defined, one of them must be class A.

VM Batch Facility Action: If the problem occurs while the VM Batch Facility is starting, it shuts down. It must have at least one job class defined in order to start.

If the problem occurs after a BATCH CONTROL SET command, the VM Batch Facility will continue to operate using either job class A or job class AUTOLOG as the only job class. CLASS AUTOLOG is used only if it was the only job class that was defined when the BATCH CONTROL SET command was issued. In all other cases, the VM Batch Facility will continue operating using job class A.

User Response: Edit the CONTROL FILE and ensure that:

- There is at least one job class defined, and all CLASS entries present are valid.

- If there is only one job class, it is job class A or job class AUTOLOG.

- If there are multiple job classes, one of them is job class A.

Then restart the VM Batch Facility or re-issue the BATCH CONTROL SET command.

DGRINI012E Error on FSREAD command on CONTROL FILE, rc= *rc*

Explanation: The VM Batch Facility was unable to read the CONTROL FILE.

VM Batch Facility Action: If the error occurred while the VM Batch Facility was starting, it has shut down.

If the error occurred after a BATCH CONTROL SET command was issued, the VM Batch Facility is continuing to operate. The CONTROL FILE keyword entries that are in effect may be:

- The entries that were in effect before the BATCH CONTROL SET command was issued. This will be the case if the FSREAD error occurred before any of the new entries had been read.
- A mix of new and old entries, if the FSREAD error occurred after some of the new CONTROL FILE entries have been read.

User Response: Look up the return code in the message in the description of the FSREAD macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility or re-issue the BATCH CONTROL SET command.

DGRINI013W The MAXIMUM-USERS keyword is missing or invalid

Explanation: The VM Batch Facility CONTROL FILE does not have a valid MAXIMUM-USERS keyword. The problem is one of these:

1. The MAXIMUM-USERS keyword is missing from the CONTROL FILE.

2. The value of the keyword is missing or invalid. For example, the value may have letters or special characters in it. It must be numeric.

The MAXIMUM-USERS keyword specifies how many job owners may have jobs in process in the VM Batch Facility at one time.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has used the default MAXIMUM-USERS keyword and value as its current MAXIMUM-USERS limit. The default keyword is shown in "Predefined Keyword Settings" on page 20.

User Response: Edit the CONTROL FILE and add the MAXIMUM-USERS keyword or change its value. Stop and restart the VM Batch Facility to put the change into effect.

DGRINI014W The MAXIMUM-JOBS keyword is missing or invalid

Explanation: The CONTROL FILE does not have a valid MAXIMUM-JOBS keyword. The problem is one of these:

1. The MAXIMUM-JOBS keyword is missing from the CONTROL FILE.

2. The value on the keyword is missing or invalid. For example, the value may have letters or special characters in it. It must be numeric.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has used the

default MAXIMUM-JOBS keyword and value as its current MAXIMUM-JOBS limit. The default keyword is shown in "Predefined Keyword Settings" on page 20.

User Response: Edit the CONTROL FILE and add a MAXIMUM-JOBS entry or change the value of the entry. Stop and restart the VM Batch Facility to put the change into effect.

DGRINI018E Insufficient virtual storage in monitor machine

Explanation: The monitor virtual machine does not have enough free virtual storage for the VM Batch Facility control program to start and run.

VM Batch Facility Action: The VM Batch Facility has shut down. This message is issued if the total virtual storage of the monitor machine is insufficient for VM Batch Facility operation.

User Response: The monitor machine must have a minimum of 2M of virtual storage. (See page 16 for a sample CP directory entry for the monitor virtual machine.)

More virtual storage may be required if your installation has raised the MAXIMUM-JOBS or MAXIMUM-USERS limits. See "Tailoring the Monitor Machine to Handle Higher Work Loads" on page 45 for help in determining the virtual storage requirements of the monitor machine if you have increased these limits.

DGRINI019I Please ignore any XXXLNK022e or XXXLMN022e messages

Explanation: The VM Batch Facility has begun to process the control file. In processing the CONTROL FILE, it puts the values of the customizing keywords in the file into effect. Any DMKLNK022e or DMKLNM022e (on non-XA, non-ESA systems) or HCPLNK022e or HCPLNM022e (on XA or ESA systems) that immediately follow this message should be ignored as they are generated as part of the normal procedure of processing the CONTROL FILE.

VM Batch Facility Action: Operation is proceeding normally.

User Response: None is required.

DGRINI020W Invalid CONTROL FILE record ignored: *record*

Explanation: In processing the CONTROL FILE, the VM Batch Facility found that the CONTROL FILE record following this message was invalid. The CONTROL FILE record is invalid for one of these reasons:

1. The first word in the record is not a VM Batch Facility keyword.

2. One or more of the values following the keyword is invalid. The value may, for example, have letters in it instead of numbers.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has ignored the invalid record shown following the message. The default value of the keyword will be used until you put a new CONTROL FILE into effect.

User Response: Consult the description of the keyword shown in the message in Chapter 3, "Customizing the VM Batch Facility" on page 19. Correct the keyword in the CONTROL FILE. Then stop and restart the VM Batch Facility or issue the BATCH CONTROL SET command to put the change into effect.

DGRINI022W Too many entries; ignoring the following CONTROL FILE record: *record*

Explanation: The CONTROL FILE record shown after this message is being ignored and will not be put into effect in the VM Batch Facility. The problem has occurred because too many keywords of the type shown have been defined in the CONTROL FILE. The message is issued while the VM Batch Facility is processing the CONTROL FILE.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the record shown after this message will not be put into effect in the program.

User Response: Edit the CONTROL FILE and reduce the number of entries of the keyword shown in this message. Remember that you cannot edit the CONTROL FILE while you are logged on to the monitor machine as it does not have write access to its 199 minidisk, where the CONTROL FILE is stored.

Limits on ADMIN, CLASS, and TASK entries are:

ADMIN   16 entries

CLASS   256 entries

TASK   128 entries.

Issue the BATCH CONTROL SET command to put the CONTROL FILE into effect after you have made the changes that you wish.

DGRINI023W Keyword value is too long: *record*

Explanation: The CONTROL FILE record shown after this message has one or more values that are too long. For example, a value that can be no longer than four characters may have five characters in it. The VM Batch Facility found the problem while starting or after a BATCH CONTROL SET command.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It's using the default value for the entry shown in the message. The default value will be used even if a duplicate entry, later in the CONTROL FILE, is valid.

If the entry is an ADMIN, TASK, or CLASS entry, it has not been put into effect but does count toward the limit of ADMIN, TASK, or CLASS entries.

User Response: Consult the description of the keyword shown in the message in Chapter 3, "Customizing the VM Batch Facility" on page 19. Correct the keyword, and stop and restart the VM Batch Facility or issue the BATCH CONTROL SET command to put it into effect.

DGRINI024W Keyword value is missing: *record*

Explanation: The CONTROL FILE record shown after this message is missing one or more required values. For example, the virtual storage size may be missing from a CLASS keyword entry. The VM Batch Facility found the problem while starting or after a BATCH CONTROL SET command.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. The VM Batch Facility is using the default value for the entry shown in the message. The default value will be used even if a duplicate entry, later in the CONTROL FILE, is valid.

If the entry is an ADMIN, TASK, or CLASS entry, it has not been put into effect but does count toward the limit of ADMIN, TASK, or CLASS entries.

User Response: Consult the description of the keyword shown in the message in Chapter 3, "Customizing the VM Batch Facility" on page 19. Correct the keyword, and stop and restart the program or issue the BATCH CONTROL SET command to put the change into effect.

DGRINI025I VM Batch Facility is ready
          Enter #CP DISC to disconnect the monitor machine and leave the VM Batch
          Facility executing

Explanation: This message indicates that the VM Batch Facility has finished its start-up procedure and is ready for work.

VM Batch Facility Action: The VM Batch Facility is ready for commands.

User Response: Enter the #CP DISC command to disconnect the monitor machine. The VM Batch Facility control program will continue running in the Batch machine and the VM Batch Facility will be ready for work.

DGRINI027E Error on FSREAD command on BATCH USERS file, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to read the BATCH USERS file. The program must be able to read that file in order to reconstruct user control blocks and restore itself to the condition it was in when it was stopped.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSREAD macro. The macro is described in your VM system library. Correct the problem indicated by the return command.

If the return code suggests that the BATCH USERS file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor machine. If the file seems to be readable, try to start the VM Batch Facility again. Otherwise, erase the file from the monitor machine 194 minidisk and start the VM Batch Facility again.

If the restart fails, see "Starting the VM Batch Facility Control Program" on page 72 for help in cold starting the VM Batch Facility.

DGRINI029E Error on FSSTATE command on BATCH USERS file, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to find the BATCH USERS file. The most likely cause of the problem is that the monitor machine 194 minidisk has not been accessed or has not been accessed as minidisk C. The program must have the file in order to reconstruct user control blocks and restore itself to the condition it was in when it stopped.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSSTATE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility.

DGRINI030E Error on FSSTATE command on BATCH JOBS file, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to find the BATCH JOBS file. The most likely cause of the problem is that the monitor machine 194 minidisk has not been accessed or has not been accessed as minidisk C. The program must have the file in order to reconstruct job control blocks and restore itself to the condition it was in when it stopped.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSSTATE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility.

DGRINI031E Error on FSREAD command on BATCH JOBS file, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to read the BATCH JOBS file. The program must be able to read that file in order to reconstruct job control blocks and restore itself to the condition it was in when it was stopped.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSREAD macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility.

If the return code suggests that the BATCH JOBS file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor virtual machine. If the file seems to be readable, try to start the VM Batch Facility again. Otherwise, erase the BATCH USERS, BATCH JOBS, and BATCH MACHINES files from the monitor machine 194 minidisk and start the VM Batch Facility again. Also, erase the monitor machine 193

minidisk where jobs are queued, and ensure that no jobs are running when you restart the VM Batch Facility. All queued jobs must be resubmitted to be run.

DGRINI032W Duplicate ADMIN entry ignored: *record*

Explanation: The ADMIN entry shown after this message is a duplicate of another ADMIN keyword in the CONTROL FILE. ADMIN keywords identify authorized users of the VM Batch Facility.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It is ignoring the duplicate entry, though the entry does count toward the limit of 16 ADMIN entries allowed in the CONTROL FILE.

User Response: None is required. When convenient, the duplicate ADMIN keyword should be removed from the CONTROL FILE or corrected.

DGRINI033W Duplicate CLASS entry ignored: *record*

Explanation: The CLASS entry shown after this message is a duplicate of another CLASS keyword in the CONTROL FILE. CLASS keywords define VM Batch Facility job classes.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It is ignoring the duplicate entry, though the entry does count toward the limit of 256 CLASS entries allowed in the CONTROL FILE.

User Response: None is required. When convenient, the duplicate CLASS keyword should be removed from the CONTROL FILE or corrected.

DGRINI034W Duplicate CONTROL FILE entry ignored: *record*

Explanation: The CONTROL FILE entry shown after this message is a duplicate of another entry.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It is ignoring the duplicate entry. If the entry is a TASK entry, it does count toward the limit of 128 TASK entries allowed in the CONTROL FILE.

User Response: None is required. When convenient, the duplicate keyword should be removed from the CONTROL FILE.

DGRINI036W MAXIMUM-USERS, MAXIMUM-JOBS cannot be changed with BATCH CONTROL

Explanation: After a BATCH CONTROL SET command, the VM Batch Facility found that the values of the MAXIMUM-USERS or MAXIMUM-JOBS keywords have been changed. The program cannot make changes to these keywords except when it is starting, because the changes require alterations to the BATCH USERS and BATCH JOBS control blocks and control block storage files.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. The new values for MAXIMUM-USERS and/or MAXIMUM-JOBS are not in effect, and the program is continuing to use the previous values of the keywords.

User Response: See "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41 for help in changing the values of the MAXIMUM-USERS and MAXIMUM-JOBS keywords.

DGRINI037W Userid not defined in system, ADMIN record ignored: *record*

Explanation: An ADMIN entry in the Control File specifies a local user ID that does not exist on this system.

In particular, the first ADMIN entry also defines the Facility alertee, and that user must be on the same node as the monitor.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the ADMIN keyword shown after the message has not been put into effect. (Exception: the VM Batch Facility will shut down if there is only one ADMIN entry in the CONTROL FILE.)

User Response: Check that you have spelled the user ID correctly and that the user ID is defined in the system's CP directory. Edit the CONTROL FILE to correct the ADMIN keyword. (Remember that you should not edit the CONTROL FILE from the monitor machine, if this error has occurred during startup, as it does not have WRITE access to its 199 minidisk, where the CONTROL FILE is stored.)

When the correction has been made, issue the BATCH CONTROL SET command or stop and restart the VM Batch Facility to put the change into effect.

DGRINI040W Error accessing 195 (accounting) minidisk

Explanation: The VM Batch Facility is unable to store accounting information on the monitor machine 195 minidisk. The problem may be that the minidisk has not been accessed or has not been accessed as the E minidisk. Or the minidisk may be too full to accept any more accounting information.

This error can occur only when the ACCOUNT keyword is set so that the VM Batch Facility accounting exit is in use.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It is temporarily working with the ACCOUNT keyword set to 0. No accounting will be done by the accounting exit.

User Response: Find out why the error has occurred. Then, after the cause of the error has been corrected, stop and restart the program or enter the BATCH CONTROL SET command. These actions will reset the ACCOUNT keyword to the value you have specified in the CONTROL FILE.

This error may have occurred because the monitor machine 195 minidisk is full or is not accessed correctly. To check for this problem:

1. Use the BATCH STOP NOW command to stop the monitor machine.

2. Log on to the monitor machine.

3. Enter this command to see if the 195 minidisk has been accessed as the E minidisk, and if it is more than 85% full:

   `query disk`

4. If the minidisk is not accessed correctly, reaccess it as the E minidisk.

5. If the minidisk is full, delete old accounting records.

6. When the problem has been corrected, restart the VM Batch Facility with this command:

   `runbatch`

7. Then enter this command to leave the monitor machine disconnected with the VM Batch Facility running:

   `#cp disc`

DGRINI042E Error on FSWRITE command on BATCH JOBS file, rc = *rc*

Explanation: While starting, the VM Batch Facility was unable to write to the BATCH JOBS file. The program must be able to write to that file in order to reconstruct job control blocks and restore itself to the condition it was in when it was stopped.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSWRITE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility.

The problem is likely to be one of these:

1. The monitor machine 194 minidisk has been accessed in read/only (R/O) mode. Correct the problem so that the minidisk is accessed in read/write (R/W) mode.

2. The monitor machine 194 minidisk is full. Correct this problem by adding more space to the minidisk.

3. The virtual storage capacity of the monitor virtual machine has been exceeded. Correct this problem by allocating more virtual storage to the monitor machine in its CP directory entry.

DGRINI044E Error on FSREAD command on BATCH MACHINES file, rc = *rc*

Explanation: The VM Batch Facility was unable to read the BATCH MACHINES file while starting or after a BATCH CONTROL SET command. The program uses the file to restore task machine control blocks.

VM Batch Facility Action: If the error occurred while the VM Batch Facility was starting, it has shut down. It needs the file to restore itself to the condition it was in when it was stopped.

If the error occurred after a BATCH CONTROL SET command, the VM Batch Facility is continuing to operate. It is using the current set of task machine control blocks in virtual storage. However, the message may indicate that the VM Batch Facility will not be able to update the BATCH MACHINES file, in which the control blocks are stored.

Also, processing of the new CONTROL FILE is likely to have been interrupted, and a mixture of new and old CONTROL FILE values may be in effect in the VM Batch Facility.

User Response: Look up the return code in the message in the description of the FSREAD macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the program or re-issue the BATCH CONTROL SET command.

If the return code suggests that the BATCH MACHINES file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor virtual machine. If the file seems to be readable, try to start the program again. Otherwise, erase the BATCH MACHINES file from the monitor machine's 194 minidisk and start the VM Batch Facility again.

If the restart fails, cold start the VM Batch Facility. See "Starting the VM Batch Facility Control Program" on page 72 for more information.

DGRINI048W Start window is invalid, TASK record ignored: *record*

Explanation: The TASK entry shown after this message has an invalid value or values in the start window times in the entry. The problem may be that:

1. There are letters or other characters in the values instead of only numbers.
2. The first two numbers are not in this range: 00–24.
3. The second two numbers are not in this range: 00–59.
4. The number as a whole is greater than 2400.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, it is not using the task virtual machine identified by the record shown in the message.

User Response: Edit the CONTROL FILE on the monitor machine 199 (G) minidisk and correct the start window time or times. The start window times are the last two values in the TASK entry.

Then put the corrected entry into effect by entering the BATCH CONTROL SET command.

DGRINI049E Error on CMS IDENTIFY command, rc = *rc*

Explanation: An error occurred when the monitor machine issued the CMS IDENTIFY command while starting. The error may indicate that there is a problem with your system.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the IDENTIFY command. The command is described in your VM system library.

Correct the problem indicated by the return code. Then restart the VM Batch Facility.

DGRINI052I Number of administrators = *nn*
    Number of classes defined = *nnn*
    Number of task machines defined = *nnn*

Explanation: This message, issued at the completion of CONTROL FILE processing during startup or after a BATCH CONTROL SET command has been issued, tells you how many administrators, classes, and task machines have been successfully defined to the VM Batch Facility. The message is sent to the monitor machine's console and is recorded in the console log file, if console logging is in effect.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

DGRINI053W Invalid value, TASK record ignored: *record*

Explanation: There is an invalid value in the TASK record which appears after this message.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, it is not using the task machine identified by the incorrect TASK entry. The TASK entry does count, however, toward the limit of 128 TASK entries allowed in the CONTROL FILE.

User Response: Edit the CONTROL FILE and check that all of the values in the TASK entry:

- Are present.
- Are of the correct type (for example, numeric values cannot have letters in them).
- Are of the correct length.

If the task machine can accept type 1 job classes (identified by a single character), a single period should precede the list of classes. There should not be any blank spaces in the list of type 1 classes.

If the task machine can accept type 2 job classes, be sure that the classid does not start with a period.

Ensure that no logon passwords were specified as '*NOPASS*' if the monitor accesses a VMBATCH DEFAULTS file specifying that logon passwords are mandatory. Ensure that a directory password was specified when a logon password was given as '*NOPASS*'.

See "Filling Out the TASK Keyword" on page 31 for more information on the TASK keyword.

When you have corrected the error in the TASK keyword, enter the BATCH CONTROL SET command to put the entry into effect.

DGRINI054W Invalid value, CLASS record ignored: *record*

Explanation: There is an invalid value in the CLASS record which appears after this message. The problem may be with any of the values that appear after the keyword.

One cause of the problem is that a default CPU seconds, print, or punch value is greater than the maximum value defined for the job class. Default values must be equal to or less than the maximum values.

VM Batch Facility Action: The VM Batch Facility continues to operate. However, it is not using the job class defined by the incorrect CLASS entry. The entry does count, however, toward the limit of 256 job class entries allowed in the CONTROL FILE.

User Response: Edit the CONTROL FILE and check for these problems with the CLASS entry:

1. Incorrect characters in a value which requires only letters or numbers.
2. Value(s) that are too long or too short.
3. Missing values.

See "Adding and Deleting Job Classes" on page 34 for more information on the CLASS keyword.

Correct the keyword. Enter the BATCH CONTROL SET command to put the correction into effect in the VM Batch Facility.

DGRINI055E Error on FSWRITE command on BATCH SYSTEM file, rc = *rc*

Explanation: The VM Batch Facility was unable to write to the BATCH SYSTEM file while starting or after a BATCH CONTROL SET command. The BATCH SYSTEM file stores information on some CONTROL FILE entries, as well as other information.

VM Batch Facility Action: If the error occurred while the VM Batch Facility was starting, it has shut down.

If the error occurred after a BATCH CONTROL SET command, the program is continuing to operate and is using the keyword values from the new CONTROL FILE. The program has not, however, updated the BATCH SYSTEM file.

User Response: Look up the return code in the message in the description of the FSWRITE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the program or re-issue the BATCH CONTROL SET command.

The problem is likely to be one of these:

1. The monitor machine's 194 minidisk has been accessed in read-only (R/O) mode. Correct the problem so that the minidisk is accessed in read-write (R/W) mode.

2. The monitor machine's 194 minidisk is full. Correct this problem by adding more space to the minidisk.

3. The virtual storage capacity of the monitor virtual machine has been exceeded. Correct this problem by allocating more virtual storage to the monitor machine in its CP directory entry.

DGRINI056E Error on FSWRITE command on BATCH MACHINES file, rc = *rc*

Explanation: The VM Batch Facility was unable to write to the BATCH MACHINES file while starting or after a BATCH CONTROL SET command. The BATCH MACHINES file stores the task machine control blocks that the program uses to keep track of task machine status and activity.

VM Batch Facility Action: If the error occurred while the VM Batch Facility was starting, it has shut down.

If the error occurred after a BATCH CONTROL SET command, the program is continuing to operate using the task machine blocks in virtual storage. The message indicates, however,

that the program may not be able to update the BATCH MACHINES file, and may not restart properly if it is stopped.

User Response: Look up the return code in the message in the description of the FSWRITE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the program or re-issue the BATCH CONTROL SET command.

The problem is likely to be one of these:

1. The monitor machine's 194 minidisk has been accessed in read/only (R/O) mode. Correct the problem so that the minidisk is accessed in read/write (R/W) mode.

2. The monitor machine's 194 minidisk is full. Correct this problem by adding more space to the minidisk.

3. The virtual storage capacity of the monitor virtual machine has been exceeded. Correct this problem by allocating more virtual storage to the monitor machine in its CP directory entry.

DGRINI057E Error on FSWRITE command on BATCH USERS file, rc = *rc*

Explanation: The VM Batch Facility was unable to write to the BATCH USERS file while starting. The BATCH USERS file stores the control blocks that keep track of job owners.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code in the message in the description of the FSWRITE macro. The macro is described in your VM system library. Correct the problem indicated by the return code. Then restart the VM Batch Facility.

The problem is likely to be one of these:

1. The monitor machine's 194 minidisk has been accessed in read/only (R/O) mode. Correct the problem so that the minidisk is accessed in read/write (R/W) mode.

2. The monitor machine's 194 minidisk is full. Correct this problem by adding more space to the minidisk.

3. The virtual storage capacity of the monitor virtual machine has been exceeded. Correct this problem by allocating more virtual storage to the monitor machine in its CP directory entry.

DGRINI058W Setting MAXIMUM-USERS to number of users still active

Explanation: While starting, the VM Batch Facility found that:

1. The MAXIMUM-USERS value in the CONTROL FILE has been reduced.
2. The number of users who have queued or running jobs is greater than the new value of MAXIMUM-USERS.

VM Batch Facility Action: The VM Batch Facility is using the current number of users as a temporary value for MAXIMUM-USERS. It has not put the value specified in the CONTROL FILE into effect.

User Response: When convenient, stop the VM Batch Facility and reduce the number of active users to the limit that you want to have in effect. Then restart the VM Batch Facility.

For help in putting changes to MAXIMUM-USERS into effect, see "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41.

DGRINI059W Setting MAXIMUM-JOBS to number of jobs still active

Explanation: While starting, the VM Batch Facility found that:

1. The MAXIMUM-JOBS value in the CONTROL FILE has been reduced.
2. The number of queued and running jobs is greater than the new value of MAXIMUM-JOBS.

VM Batch Facility Action: The VM Batch Facility is using the current number of jobs as a temporary value for MAXIMUM-JOBS. It has not put the value specified in the CONTROL FILE into effect.

User Response: When convenient, stop the VM Batch Facility and reduce the number of active jobs to the limit that you want to have in effect. Then restart the VM Batch Facility.

For help in putting changes to MAXIMUM-JOBS into effect, see "Changing MAXIMUM-USERS and MAXIMUM-JOBS Limits" on page 41.

DGRINI5nn

User Response: See "Messages Numbered 500 and Higher" on page 224

DGRMAI002W Invalid return code from user exit *exit***, rc =** *rc*

Explanation: The VM Batch Facility has received an invalid return code from an exit. Only the return codes documented in the prologs of the exits can be accepted by he program.

VM Batch Facility Action: The VM Batch Facility has ignored the invalid return code. It has acted as if the exit issued return code 0.

User Response: If you have modified an exit, correct the exit so that only the documented return codes are used.

DGRMAI003E VM Batch Facility stopping; error in user exit DGRUAC, rc = *rc*

Explanation: The job accounting exit, DGRUAC EXEC, has returned a nonzero return code to the monitor machine. The return code in the message is the nonzero return code of a failed CP or CMS command issued by DGRUAC.

The VM Batch Facility interprets nonzero return codes from DGRUAC as a warning that a serious error has occurred in DGRUAC EXEC.

DGRUAC EXEC sends a message to the alert message receiver indicating which command has failed. (If the exit is not able to send a message to the alert message receiver, it will send the message to the monitor machine console, where it will be recorded if console message spooling is active.)

VM Batch Facility Action: The VM Batch Facility monitor machine is stopping. Task machines will finish running their current jobs and will then log off.

Accounting data for currently running jobs that are completed while the monitor is logged off will not be complete. The accounting records will reflect the last resource usage data available to the monitor machine before it shut down. The records are created when the VM Batch Facility is restarted.

User Response: Look at the separate message sent by the DGRUAC EXEC to indicate which command has failed. The message is sent to the alert message receiver or to the monitor machine console, if the alert receiver is not receiving messages. Look up the command and return code shown in the DGRUAC EXEC message in your VM system library. When you have corrected the problem indicated by the return code, restart the VM Batch Facility using the procedure shown in the *VM Batch Facility Installation, Customization, and Administration* manual.

Note that some job accounting data may be missing or in error. DGRUAC job accounting data is kept in the DGRUAC RECORDS file on the monitor machine 195 minidisk.

DGRMAI004W Error erasing job *userid nodeid jobid***, FSERASE rc =** *rc*

Explanation: The VM Batch Facility was unable to erase the job file named in the message. The FSERASE macro, issued to erase the job file, failed with the return code shown in the message.

The program erases job files from the 193 minidisk, where jobs are queued, after jobs are completed or canceled.

VM Batch Facility Action: None. The VM Batch Facility continues normal operation.

User Response: None is required unless the error is repeated. If repeated failures occur, look up the FSERASE return code in your VM system library. Correct the error. Also, check to see if the job files are still stored on the monitor machine 193 minidisk the next time that the VM Batch Facility is stopped.

Leaving job files named in these messages on the 193 minidisk will usually not cause errors or trouble for users of the VM Batch Facility. However, the files use up space on the minidisk.

DGRMAI006W VM Batch Facility job *userid nodeid jobid* **is stalled**

Explanation: A problem may have occurred with the job shown in the message. The job was started but apparently has stopped running and may be stalled. The job may have stalled because an error in the job exec has stopped it from running. Or the job may have stalled temporarily because the system work load is high, for example.

VM Batch Facility Action: The task machine running the stalled job is logged on but execution of the job apparently has stopped. If the job does not restart within the time and stall warning limits set by your installation, the DGRIDO exit will be called. The default action by this exit is to have the job canceled.

User Response: Take any of these actions:

1. Let the situation be resolved by the DGRIDO exit. If it is set up to cancel the job, this will occur automatically if the job has not restarted within a few minutes.

2. Cancel the job yourself. Canceling the job will cause the VM Batch Facility to discard it right away.

3. Prevent the VM Batch Facility from canceling the job by suspending the task machine and job. You must be a VM Batch Facility authorized user to suspend a job.

   To suspend a job and keep its task machine logged on indefinitely, issue these commands:

   a. BATCH STATUS *userid nodeid jobid*–to find out which task machine is running the job.

   b. BATCH SUSPEND *taskid*–to suspend the task machine running the job.

Before submitting the job again, check to see that it does not use prohibited commands. Refer to the *VM Batch Facility User's Guide* for information on commands that cannot be issued from job execs.

DGRMAI007W Job *userid nodeid jobid* **canceled by break in job chain**

Explanation: The job named in this message has been canceled for one of these reasons:

- The job to which this job was chained has been canceled.
- The job to which this job was chained ended before reaching the jobstep specified in the CHAIN option for this job.

A chained job is one whose eligibility to start depends on the progress of another job. Job chaining is set up with the CHAIN option of the BATCH SUBMIT and BATCH CHJOB commands.

VM Batch Facility Action: The VM Batch Facility has canceled the job named in the message.

User Response: None is required. If you wish to resubmit the chained jobs, first determine why the first job in the chain was canceled. The console log produced by the canceled job may help you to determine why the job was canceled. The console log file is sent to the owner of the job when the job is canceled.

After you have corrected the problem with the first job, resubmit the jobs, specifying the job chaining that you require.

DGRMAI008W VM Batch Facility could not purge job *userid nodeid jobid* **spoolid** *nnnn*
        from punch, CP PURGE rc = *rc*

Explanation: The VM Batch Facility has encountered a problem with the monitor machine punch. The CP PURGE PUNCH command that the program issued has failed with the return code shown in the message.

The job identified in the message has been started or canceled, and the program was attempting to purge the job file from the monitor machine punch. Since the CP PURGE command failed, the job file may still be queued in the monitor machine punch.

The job file identified in the message was held in the punch because it would not fit on the 193 minidisk, where jobs normally are held.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, it may not be able to transfer job files from its punch to task machines until the problem is corrected.

User Response: Look up the return code shown in the message in the description of the CP PURGE command and correct the problem indicated by the return code. The command is documented in your VM system library.

DGRMAI010W Unable to write job *userid nodeid jobid* **block, CMS FSWRITE rc =** *rc*

Explanation: The VM Batch Facility encountered a problem when it tried to write data from the virtual job control block named in the message to the BATCH JOBS file. The FSWRITE macro issued by the program failed with the return code shown in the message.

The program uses job blocks to keep track of jobs in process. As job status changes, the program updates the information in the job blocks. The blocks are stored in the BATCH JOBS file on the monitor machine 194 minidisk.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the message may indicate that the program is unable to update the BATCH JOBS file. It may not be able to restart correctly if it is stopped.

User Response: Look up the return code shown in the message in the description of the CMS FSWRITE macro. The command is described in your VM system library. When you have corrected the problem, the program should resume normal operation.

Note: If the return code suggests that the file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor virtual machine. If the file seems to be readable, try to start the VM Batch Facility again. Otherwise, erase the file, together with the BATCH USERS and BATCH MACHINES files, from the monitor machine 194 minidisk and start the VM Batch Facility again. Also, erase the monitor machine's 193 minidisk, where jobs are queued. All queued jobs must be resubmitted to be run.

DGRMAI011W Unable to write task *userid* **block, CMS FSWRITE rc =** *rc*

Explanation: The VM Batch Facility encountered a problem when it tried to store data in the task control block named in the message. The FSWRITE macro issued by the program failed with the return code shown in the message.

The program uses task blocks to keep track of task virtual machines. As task machine status changes, the program updates the information in the blocks. The blocks are stored in the BATCH MACHINES file on the monitor machine 194 minidisk. The blocks are named by the user IDs of the task machines.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the message may indicate that the program is not able to update the BATCH MACHINES file. It may not restart correctly if it is stopped.

User Response: Look up the return code shown in the message in the description of the CMS FSWRITE macro. The macro is described in your VM system library. When you have corrected the problem, the program should resume normal operation.

Note: If the return code suggests that the BATCH MACHINES file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor virtual machine. If the file seems to be readable, try to start the VM Batch Facility again. Otherwise, erase the BATCH MACHINES file from the monitor machine 194 minidisk and start the VM Batch Facility again.

If the restart fails, cold start the VM Batch Facility. See "Starting the VM Batch Facility Control Program" on page 72 for more information.

DGRMAI012W Unable to write user *userid nodeid* **block, CMS FSWRITE rc =** *rc*

Explanation: The VM Batch Facility encountered a problem when it tried to store data in the user block named in the message. The FSWRITE macro issued by the program failed with the return code shown in the message.

The program uses user blocks to keep track of job owners. As job owner status changes, the program updates the information in the blocks. The blocks are stored in the BATCH USER file on the monitor machine 194 minidisk. The blocks are named by the user IDs and node IDs of job owners.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the message may indicate that the program is unable to update the BATCH USERS file. It may not be able to restart if it is stopped.

User Response: Look up the return code shown in the message in the description of the CMS FSWRITE macro. The macro is described in your VM system library. When you have corrected the problem, the program should resume normal operation.

Note: If the return code suggests that the file has been made unreadable, check the file by editing it. It is located on the 194 (C) minidisk of the monitor virtual machine. If the file seems to be readable, try to start the VM Batch Facility again. Otherwise, erase the file and start the VM Batch Facility again.

If the restart fails, cold start the VM Batch Facility. See "Starting the VM Batch Facility Control Program" on page 72 for more

---

DGRMAI018W Error routine canceled for job *userid nodeid jobid*

Explanation: Processing of the ERROR job control option for the specified job has been terminated. ERROR option processing is done when a job has been canceled.

The ERROR option processing has been canceled for one of these reasons:

- ERROR option processing exceeded the Print Records (PRINT), Punch Records (PUNCH), or CPU Seconds (SECONDS) limit in effect for it.
- ERROR option processing stalled and was canceled.
- Another event occurred to cause ERROR option processing to be canceled. For example, an ERROR option exec is canceled if it executes a LOGOFF command.

VM Batch Facility Action: Both the original job and ERROR option processing have been stopped. The VM Batch Facility will send the ERROR option console log to you.

User Response: None is required.

DGRMAI5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRMSG001E Error during message processing. Probable cause is insufficient virtual storage

Explanation: This message indicates that the monitor machine does not have enough virtual storage available for message processing. Messages are not being screened by the message screening exit. This message is sent to the monitor machine's console only.

VM Batch Facility Action: The VM Batch Facility is continuing to send out messages as required. It is not, however, calling the DGRUMSG exit to screen messages before they are sent.

User Response: Increase the virtual storage of the monitor virtual machine.

DGRMSG5nn

User Response: See "Messages Numbered 500 and Higher" on page 224

DGRRDR001I *command* **from** *userid* **at** *nodeid* **[for** *userid* **at** *nodeid* **]**

Explanation: Records each command received by the VM Batch Facility from users. The message is displayed at the monitor machine's console if you are logged on to it. It is also recorded in the VM Batch Facility console message log, if console message spooling is active.

VM Batch Facility Action: The message indicates normal operation.

User Response: None is required.

DGRRDR002E Job/Chjob *userid nodeid jobid* **rejected, invalid option:** *option*

Explanation: The VM Batch Facility is responding to a BATCH SUBMIT or BATCH CHJOB command you entered. The program has rejected the command because one or more of the command options or panel fields you used was incorrect.

VM Batch Facility Action: The VM Batch Facility has not carried out the command. It is ready for another command.

User Response: Consult the *VM Batch Facility User's Guide* for information about SUBMIT, or "BATCH CHJOB: Changing Job Control Options" on page 88. Correct the error in the command or panel input and enter the command again.

If you entered the command yourself, the problem may be that you misspelled an option or entered an incorrect value with the option.

If you entered the command from a panel, the problem is with one of the values you entered on the panel.

DGRRDR003E Invalid spool file received; spoolid *nnnn* **purged**

Explanation: The monitor machine has received a spool file that was neither a command nor a job file. The VM Batch Facility has ignored the spool file and has purged it from its reader.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required.

DGRRDR004E Error reading spool file *nnnn***, DIAGNOSE X'14' rc =** *rc*

Explanation: The VM Batch Facility was unable to use the DIAGNOSE X'14' instruction to read a spool file–a command or job. The DIAGNOSE X'14' failed with the error code shown in the message. The problem may have occurred because the system has a high work load.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has gone on to try to read other spool files. The spool file that the program could not read remains in the

reader. The program will attempt to read the spool file again. However, the command or job will remain unexecuted or unqueued until the file has been read.

User Response: Look up the error code in the description of the DIAGNOSE X'14' instruction in your VM system library. The problem may be transient and may not require correction.

DGRRDR005I Processing file *spoolid* **from** *userid* **at** *nodeid*

Explanation: Records every spool file received in the monitor virtual machine reader. The message is recorded in the VM Batch Facility console message log to help audit VM Batch Facility activity, if console message logging is active.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required.

DGRRDR006E Error on CMS RDCARD macro, rc = *rc*

Explanation: The VM Batch Facility encountered a problem while reading a spool file from the BATCH user ID reader. The CMS RDCARD macro issued by the VM Batch Facility failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, it has discarded the file which it was unable to read.

User Response: Look up the return code shown in the message in the description of the RDCARD macro. The macro is described in your VM system library. Take corrective action as required.

DGRRDR009E Not permitted to use VM Batch Facility authorized user commands

Explanation: You entered a VM Batch Facility authorized user command, but you have not been identified as an authorized user. If you are not an authorized user, you can use only the SUBMIT, CHJOB, PASSWORD, QUERY, and JOBSTEP commands, and the nonprivileged forms of the CHJOB, STATUS, and CANCEL commands.

VM Batch Facility Action: Command not performed. The VM Batch Facility is ready for another command.

User Response: Refer to the *VM Batch Facility User's Guide* for descriptions of the commands you can use.

DGRRDR010I Job *userid nodeid jobid* **not found**

Explanation: The VM Batch Facility has received a BATCH CHJOB command, but could not find the job that was to have been changed. The job may have already been completed or canceled.

This message may also be issued in response to a job submission when no job EXEC was included in the job deck.

This message is sent to the command issuer and to the monitor machine's console for recording in the console message log file.

VM Batch Facility Action: The VM Batch Facility has ignored the command and is continuing to operate.

User Response: None is required.

DGRRDR014I Command JOBSTEP *option* **for job** *userid nodeid jobid*

Explanation: Records the reception of a BATCH JOBSTEP command from a task machine. The message is sent only to the monitor machine's console for recording in the console message log file.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required.

DGRRDR015I Command JOBEND *jobend status* **for job** *userid nodeid jobid*

Explanation: Records the reception of a JOBEND message from a task machine. JOBEND indicates that the job running in the task machine has completed or canceled.

The message is sent only to the monitor machine's console for recording in the console message log file.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required.

DGRRDR017E Job *userid nodeid jobid* **rejected: duplicate jobid**

Explanation: The job identified in the message has been rejected because another of your jobs that is currently in process has the same job ID.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Submit the job again, specifying a different job ID for the job.

DGRRDR019I Job *userid nodeid jobid* **rejected: more than 100 records received before *END.OF.JOB.PARAMETERS**

Explanation: The job you submitted has been rejected; the VM Batch Facility read more than 100 cards or records in the BATCH SUBMIT command before reaching the *END.OF.JOB.PARAMETERS record.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Check that your BATCH SUBMIT command does not contain more than 100 records. Then submit the job again.

DGRRDR021W CPU seconds limit too high, changed to job class maximum

Explanation: The CPU Seconds (SECONDS job control option) limit you specified for the job you just submitted exceeded the job class limit on CPU seconds. Jobs are not allowed to exceed the class limit, and the Seconds limit on your job has been reset to the maximum value defined for the job class.

VM Batch Facility Action: The VM Batch Facility has accepted your job and has changed the CPU Seconds limit for the job.

User Response: To review the CPU seconds limit of the job class to which your job belongs, use the Query panel or enter the BATCH QUERY CLASS command. If you did not specify a job class when you submitted your job, it has been assigned to job class A.

If the CPU Seconds limit of the job class is not high enough, use the Chjob panel or BATCH CHJOB command to switch your job to a job class that has an adequate limit.

DGRRDR022W Print records limit too high, changed to job class maximum

Explanation: The Print Records (PRINT job control option) limit you specified in a BATCH SUBMIT or BATCH CHJOB command exceeded the job class limit on Print Records. Jobs cannot exceed the class limit, and the Print limit on your job has been reset to the maximum value defined for the job class.

VM Batch Facility Action: The VM Batch Facility has accepted your job and has changed the Print Records limit for the job.

User Response: To review the Print Records limit of the job class to which your job belongs, use the Query panel or enter the BATCH QUERY CLASS command. If you did not specify a job class when you submitted your job, it has been assigned to job class A.

If the Print Records limit of the job class is not high enough, use the Chjob panel or BATCH CHJOB command to switch your job to a job class that has an adequate limit.

DGRRDR023W Punch records limit too high, changed to job class maximum

Explanation: The Punch Records (PUNCH job control option) limit you specified for a BATCH SUBMIT or BATCH CHJOB command exceeded the job class limit on Punch Records. Jobs cannot exceed the class limit, and the Punch limit on your job has been reset to the maximum value defined for the job class.

VM Batch Facility Action: The VM Batch Facility has accepted your job and has changed the Punch Records limit for the job.

User Response: To review the Punch Records limit of the job class to which your job belongs, use the Query panel or enter the BATCH QUERY CLASS command. If you did not specify a job class when you submitted your job, it has been assigned to job class A.

If the Punch Records limit of the job class is not high enough, use the Chjob panel or BATCH CHJOB command to switch your job to a job class that has an adequate limit.

DGRRDR026E Job/Chjob *userid nodeid jobid* **rejected; cannot be chained to itself**

Explanation: Your job or CHJOB command has been rejected because the Chain to Job field or CHAIN option attempted to chain the job to itself. Jobs can be chained only to other jobs.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Resubmit the job or enter the CHJOB command again. Change the job chaining so that the start of the job depends on the progress of another job.

DGRRDR027E Job/Chjob *userid nodeid jobid* **rejected; could not be chained**

Explanation: Your job or CHJOB command has been rejected; the VM Batch Facility could not carry out the job chaining that you specified for the job.

The job chaining could not be satisfied for one of these reasons:

1. The other job may already have been completed or canceled or had its final status record erased.
2. The other job may not have been submitted yet.
3. The other job may have been a CLASS AUTOLOG job. Jobs cannot be chained to CLASS AUTOLOG jobs.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Resubmit the job or enter the CHJOB command again. Check that you have correctly identified the job to which you want to chain this job.

You can review the job IDs of your own jobs with the BATCH STATUS command. To review jobs belonging to other users, issue the authorized user BATCH STATUS *userid* ALL command, or have a VM Batch Facility authorized user help you.

DGRRDR028E Job/Chjob *userid nodeid jobid* **rejected; job class not defined**

Explanation: Your job or CHJOB command has been rejected because the job class to which you assigned the job is not known to the VM Batch Facility. Job classes must be defined to the program in the CONTROL FILE.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Submit the job again, or reenter the CHJOB command. Check that you spelled the job class name correctly. You can review the VM Batch Facility job classes by using the Query panel or by entering the BATCH QUERY CLASS command.

If the job class is missing and you believe that it should be defined, contact the VM Batch Facility administrator.

DGRRDR029E Job *userid nodeid jobid* **rejected by user exit DGRUJB**

Explanation: The job you submitted has been rejected because the job screening exit would not allow the VM Batch Facility to accept the job. The job screening exit (DGRUJB EXEC) can be changed to reject or accept jobs based on your installation's own criteria. When the program is first installed, the job screening exit automatically rejects all jobs sent from Remote Job Entry (RJE) workstations.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Contact the VM Batch Facility administrator to determine why your job was not allowed to run. You may also be able to review the DGRUJB EXEC by editing the exec, which is stored on the monitor machine's 199 minidisk.

DGRRDR030E Job *userid nodeid jobid* **rejected; exceeds user job limit**

Explanation: The job you submitted has been rejected because the number of jobs you own that are queued and/or running has reached the limit set by your installation.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Submit the job again as soon as one of your other jobs has been canceled or completed. Contact the VM Batch Facility administrator if you believe that the limit on the number of jobs each user can have queued and/or running at one time should be raised.

DGRRDR031E Command *command* **rejected by user exit DGRUCD**

Explanation: The command you entered has been rejected because the command screening exit would not allow the VM Batch Facility to perform the command.

The command screening exit (DGRUCD EXEC) can be changed to reject or accept commands based on your installation's own criteria. Note that when the program is first installed, the command screening exit automatically rejects all jobs and commands sent from Remote Job Entry (RJE) workstations.

VM Batch Facility Action: The VM Batch Facility is ready for another command.

User Response: Contact the VM Batch Facility administrator to determine why your command was rejected. The exits are stored on the monitor machine's 199 minidisk.

DGRRDR033I Job *userid nodeid jobid* **queued in punch due to job size**

Explanation: The job named in the message exceeded the MAXIMUM-JOB-SIZE limit set in the CONTROL FILE. Instead of being queued on the monitor machine 193 minidisk, the file is queued in the it's punch. The job is still eligible to run and is still subject to VM Batch Facility commands.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required.

DGRRDR034W Job *userid nodeid jobid* **queued in punch; not enough 193 disk space**

Explanation: The job named in the message has been queued in the monitor machine's punch, instead of on it's 193 minidisk. The 193 minidisk, used for storing job files while jobs are queued and running, does not have enough space to store the job file. The file may exceed the MAXIMUM-JOB-SIZE limit on the size of job files, or the 193 minidisk may be full.

Though it is queued in the punch, the job is eligible to run and is subject to VM Batch Facility commands.

This message is sent to the monitor machine's console, for recording in the console message log file, and to the alert message receiver.

This message can also occur if the monitor machine's 193 minidisk is not defined, linked or accessed correctly.

VM Batch Facility Action: The VM Batch Facility is continuing to operate.

User Response: None is required. If you see this message frequently, consider increasing the size of the monitor machine's 193 minidisk. The minidisk may not be large enough to accommodate your present batch processing work load. As a result, the VM Batch Facility is using system spool space to store job files.

DGRRDR035W Error renaming *filename*, **CMS RENAME rc =** *rc*

Explanation: There was a CMS RENAME error when loading a job to the monitor 193 mini-disk.

VM Batch Facility Action: The job is queued in the punch and processed normally, as if the disk was full.

User Response: None required.

DGRRDR036E Chjob *userid nodeid jobid* **rejected; job not queued**

Explanation: The BATCH CHJOB command you entered has been rejected for these reasons:

1. The job has already started to run, and one or more of the job control options you used cannot be changed once the job has started to run.
2. The job has completed.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Enter the command again, using only options that are valid after a job has started running.

DGRRDR037E CHJOB for *userid nodeid jobid* **rejected by user exit DGRUJB**

Explanation: The VM Batch Facility has rejected the CHJOB command issued for the job named in the message. The command was rejected because the exit, DGRUJB, instructed the VM Batch Facility to do so by setting a return code.

VM Batch Facility Action: The job remains unchanged. The VM Batch Facility is ready for another command.

User Response: Check with your VM Batch Facility administrator as to the reason for the exit, DGRUJB, rejecting your command.

DGRRDR050I Received unexpected IUCV interrupt: type = *type*

Explanation: The VM Batch Facility has received an IUCV interrupt to which it could not respond. This message is sent only to the monitor machine's console, for recording in the console message log file.

VM Batch Facility Action: The VM Batch Facility has ignored the interrupt and is continuing to operate.

User Response: None is required. If the message recurs often, there may be an IUCV problem in your system. Consult your VM system library for more information on IUCV and on the type of IUCV interrupt received by the VM Batch Facility.

DGRRDR051E Error on IUCV or CMSIUCV function *function***, rc =** *rc*

Explanation: The macro named in the message failed with the return code shown in the message when issued by the monitor machine. The message has been sent to the monitor machine's console for recording in the console message log, and to the alert message receiver.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Look up the return code shown in the message in the description of the macro named in the message. The macros are described in your VM system library.

Correct the problem indicated by the return code and restart the VM Batch Facility.

DGRRDR052W Error on IUCV RECEIVE, IPRCODE = *code*

Explanation: The IUCV RECEIVE macro failed with the IPRCODE shown in the message when it was issued by the monitor machine.

The message is sent to the monitor machine console for recording in the console message log file, and to the alert message receiver.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has purged the path for which the RECEIVE was issued.

User Response: Look up the IPRCODE code shown in the message in the description of the IUCV macro named in the message. The macros are described in your VM system library. Correct the problem indicated by the return code.

DGRRDR055W Job *userid nodeid jobid* **queued in punch, FSWRITE rc=** *rc*

Explanation: The job named in the message has been queued in the BATCH *userid* punch, instead of on the BATCH 193 minidisk. An error occurred while writing the job to the 193 minidisk.

Though it is queued in the punch, the job is eligible to run and is subject to VM Batch Facility commands. This message is sent to the BATCH console, for recording in the console message log file, and to the alert message receiver.

VM Batch Facility Action: The VM Batch Facility continues to operate.

User Response: Look up the return code in the message in the description of the FSWRITE macro. The macro is described in your VM system library. Shutdown the VM BATCH Facility. Correct the problem indicated by the return code. Then restart the VM BATCH Facility. The problem is likely to be one of these:

1. The batch 193 minidisk has been accessed in read-only (r/o) mode.
2. The virtual storage capacity of the batch machine has been exceeded.
3. A permanent i/o error has occurred on the 193 minidisk.

DGRRDR5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRTAS5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRTSK002E Parameter(s) missing or invalid

Explanation: An invalid parameter has been passed to the module.

VM Batch Facility Action: The job is canceled.

User Response: Verify that the parameters are correct.

DGRTSK003E DIAG 68 RC = nn (AUTHORIZE BATCH SMSG, FAILED)

Explanation: VMCF authorization has failed with return code specified in the message.

VM Batch Facility Action: The job is canceled.

User Response: Verify that the variable batchid is specified correctly in DGRTASK exec.

DGRTSK5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRUVM002E Unable to read file *filename***, FSREAD rc =** *rc*

Explanation: The VM Batch Facility was unable to read an APNDLIST file that you specified for the job that you just submitted. The FSREAD macro issued by the VM Batch Facility failed with the return code shown in the message.

VM Batch Facility Action: Your job has not been submitted. The VM Batch Facility is ready for another command.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the administrator, refer to the description of the FSREAD macro in your VM system library. Correct the problem indicated by the return code and submit the job again.

DGRUVM003E Invalid filename in file *filename*

Explanation: The VM Batch Facility was unable to find one of the files that you listed in the APNDLIST file named in the message. The problem may be that:

 1. The file name was wrong, or the file is not located on the minidisk you specified.

 2. The file name was not shown in exactly the same way in the APNDLIST file as it is recorded on the minidisk on which it is stored.

VM Batch Facility Action: Your job has not been submitted. The VM Batch Facility is ready for another command.

User Response: Check to be sure that the file is located on the minidisk you specified in the APNDLIST file. Also, use the FILELIST command to check the name of the file and see whether it is shown in capitals or in mixed case letters. The file name must be listed in the APNDLIST file in the same way as it appears in the filelist.

Then correct the APNDLIST file and submit the job again.

DGRUVM004E File *filename* **not found, FSSTATE rc =** *rc*

Explanation: The VM Batch Facility was unable to find the job file you submitted or one of the files you listed in an APNDLIST file. The job file or missing APNDLIST file is named in the message. The FSSTATE command issued by the program to find the file failed with the return code shown in the message.

VM Batch Facility Action: Your job has not been submitted. The VM Batch Facility is ready for another command.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the administrator, refer to the description of the FSSTATE macro in your VM system library. Correct the problem indicated by the return code and submit the job again.

DGRUVM005E Error appending file *filename***, CMS DISK DUMP rc =** *rc*

Explanation: The VM Batch Facility was unable to append a job file or APNDLIST file that you have submitted. The file that could not be appended is named in the message.

The DISK DUMP command issued by the program to read the file failed with the return code shown in the message.

VM Batch Facility Action: Your job has not been submitted. The VM Batch Facility is ready for another command.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the administrator, refer to the description of the DISK DUMP command in your VM system library. Correct the problem indicated by the return code and submit the job again.

DGRUVM006E Parameter(s) missing or invalid

Explanation: The VM Batch Facility received an invalid command from a task machine or job owner. One or more of the options of the command are not known to the program.

VM Batch Facility Action: The VM Batch Facility has not carried out the command. It is continuing to operate.

User Response: If you find this message in the console log file generated by one of your jobs, check the BATCH JOBSTEP commands in your job to be sure that they are correct.

DGRUVM010I Sending JOBEND to monitor machine via spool file

Explanation: A task machine was unable to establish an IUCV connection to the monitor virtual machine after finishing a job. The task machine is sending the JOBEND message to the monitor machine by spool file instead of via IUCV.

The problem probably has occurred for one of these reasons:

1. The VM Batch Facility control program is not running in the monitor machine.

2. The BATCH monitor machine is not logged on.

This message is sent to the monitor machine and is recorded in the console log file, if console message spooling has been activated.

VM Batch Facility Action: An IUCV problem or a problem with the VM Batch Facility may exist. If the VM Batch Facility control program is running, the program will execute the JOBEND command and normal job completion processing will take place. If it is not running, job completion processing will take place, if possible, when the VM Batch Facility is restarted. Accounting data for the job may be incomplete.

User Response: None is required.

DGRUVM011E Error on CMSIUCV HNDIUCV macro, rc = *rc***,**

Explanation: The CMSIUCV HNDIUCV macro failed with the return code shown in the message. The VM Batch Facility was using the macro to establish an IUCV connection between a user's virtual machine or a task machine and the monitor virtual machine.

This message appears at one of these locations:

- In the console log file generated by the task machine that was running the job. The console log file is sent to the job owner.
- At the terminal of a user.

VM Batch Facility Action: No IUCV connection could be established between a task virtual machine or user's virtual machine and the monitor machine. The VM Batch Facility may not be running.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the VM Batch Facility administrator, look up the return code shown in the message in the description of the CMSIUCV macro named in the message. The macro is described in your VM system library. Correct the problem indicated by the return code.

DGRUVM012I Unable to transmit IUCV command to *userid*, **SEND IPRCODE =** *code*

Explanation: The IUCV SEND macro failed with the IPRCODE shown in the message when it was issued by the monitor machine.

This message appears at one of these locations:

- In the console log file generated by the task machine that was running the job. The console log file is sent to the job owner.
- At the terminal of a user.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. It has purged the job file or command that it was trying to send when the failure occurred.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the VM Batch Facility administrator, look up the IPRCODE code shown in the message in the description of the SEND macro. The macro is described in your VM system library. Correct the problem indicated by the return code.

DGRUVM015E Unable to get IUCV connection to monitor machine, CMSIUCV CONNECT rc = *rc*

Explanation: The CMSIUCV CONNECT macro failed with the return code shown in the message. The VM Batch Facility was using the CONNECT macro to get a CMSIUCV connection between a task machine or job owner's machine and the monitor virtual machine. A likely cause of the problem is that the VM Batch Facility control program, in the monitor machine, is not running.

This message appears either:

- In the console log file generated by a task machine that has issued a BATCH JOBSTEP command or a JOBEND message to the monitor machine. (The console log file is sent to the job owner.)
- At the terminal of a user.

VM Batch Facility Action: The VM Batch Facility control program may have stopped running.

User Response: Have the VM Batch Facility administrator or system administrator help you with this problem.

If you are the VM Batch Facility administrator, look up the return code shown in the message in the description of the CONNECT macro. The macro is described in your VM system library.

Correct the problem indicated by the return code. A likely cause of the problem is that the control program in the monitor machine has stopped running.

DGRUVM016E Commands not allowed from monitor machine

Explanation: The VM Batch Facility cannot execute VM Batch Facility commands that are issued from the monitor machine.

VM Batch Facility Action: The command has not been carried out.

User Response: Log on to another user ID to issue the VM Batch Facility command you wanted to use. The VM Batch Facility control program cannot accept commands that are issued from the virtual machine in which it is running.

DGRUVM018E Invalid append file specified

Explanation: An append file in the BATCH SUBMIT command that you entered had an invalid file ID. The problem may have been that the file name, file type, and/or file mode were too long or too short.

VM Batch Facility Action: The command has not been carried out. The VM Batch Facility is ready for another command.

User Response: Check the file IDs of appended files and enter the command again.

DGRUVM019E Punch *cuu* does not exist and cannot be defined, DEFINE rc = *rc*

Explanation: The following problems have occurred:

 1. The VM Batch Facility found that no punch was defined at address 00D in a job owner's virtual machine; and

 2. The VM Batch Facility was not able to define a punch at address 00D using the CP DEFINE command. The command failed with the return code shown in the message.

The VM Batch Facility requires a punch at address 00D in a user's virtual machine in order to send BATCH SUBMIT and BATCH CHJOB commands to the monitor virtual machine.

VM Batch Facility Action: The command has not been sent to the VM Batch Facility.

User Response: Look up the return code shown in the message in the description of the CP DEFINE command. The description appears in your VM system library.

When you have corrected the problem, enter the command again.

DGRUVM020I Received unexpected IUCV interrupt, type = *type*

Explanation: The DGRUVM module of the VM Batch Facility received an IUCV interrupt that it did not recognize. The type of interrupt is shown in the message.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. The command that was being transmitted has not been discarded. DGRUVM module will continue to try to pass the command to the VM Batch Facility monitor machine for execution.

User Response: None is required.

DGRUVM021E SMSG to *rscsid* failed, rc = *rc*

Explanation: When sending commands to a remote monitor, the command is communicated using SMSG to the RSCS virtual machine on your node. This message indicates a failure of the SMSG command.

VM Batch Facility Action: The command has not been communicated to the monitor you are addressing.

User Response: Check that your SYSTEM NETID file correctly identifies the name of the RSCS virtual machine on your system.

DGRUVM5nn

User Response: See "Messages Numbered 500 and Higher" on page 224.

DGRVMC001E VM Batch Facility stopping; system not supported

Explanation: The DIAGNOSE X'00' issued by the VM Batch Facility monitor machine indicates that this system is not a VM system that is supported by the VM Batch Facility.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: See the *VM Batch Facility General Information* manual or *VM Batch Facility Licensed Program Specification* for a list of VM Batch Facility software requirements.

DGRVMC002E DIAGNOSE X'00' not supported on this system

Explanation: This system does not support the DIAGNOSE X'00' instruction. The VM Batch Facility needs the instruction to get required information about the system.

VM Batch Facility Action: The VM Batch Facility has shut down.

User Response: Ensure that the VM Batch Facility has been installed in one of the VM systems it supports. See the *VM Batch Facility General Information* manual or *VM Batch Facility Licensed Program Specifications* for a list of VM Batch Facility software requirements.

DGRVMC003W Job *userid nodeid jobid* **not authorized to use ALTID** *userid*

Explanation: The alternate user ID requested could not be set due to a rc 12 received when the DIAGNOSE X'D4'. was issued. This is a CP/RACF user DIAGNOSE which implements the alternate user ID function.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the task machine will be able to link only to it's own minidisks. It will not be able to link successfully to minidisks owned by the job owner, and any spool files originating from the task machine will not have their ORIGINID set to the requested alternate userid.

User Response: Issue the required RACF PERMITs, or consult your RACF Security personnel.

DGRVMC004E Invalid password for task machine *userid*

Explanation: The logon password of the task machine identified in the message is incorrect.

The error may have occurred because the logon passwords are not the same in the CONTROL FILE and the CP directory entry of the task machine.

If the Resource Access Control Facility (RACF) is installed in your system, the error may also have occurred because a new RACF password was not recorded in the CP directory entry of the task machine.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, it was not able to log on the specified task machine and cannot use the task machine to run jobs. The task machine status has been changed to OFFLINE. The program will not try to log on the task machine until:

- The task machine is restarted with the BATCH START command; or

- The VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is issued. These two events automatically reset OFFLINE task machines.

User Response: Ensure that the logon password of the task machine is the same in all of these places:

1. The VM Batch Facility CONTROL FILE on the monitor machine's 199 minidisk.

2. The CP directory entry that defines the virtual machine.

3. In RACF, if it is installed in your system.

Correct the password as required. If the CONTROL FILE password was incorrect, or had not been put into effect in the VM Batch Facility, enter the BATCH CONTROL SET command.

When all passwords are correct, enter the BATCH START command to restart the task machine. The VM Batch Facility should be able to log on the task machine.

DGRVMC005E Task machine *userid* **not defined in system, not available**

Explanation: The VM Batch Facility tried to use the task machine identified in the message, but found that the task machine is not defined in the CP directory of the system. The task machine is defined in the CONTROL FILE.

VM Batch Facility Action: The VM Batch Facility has marked the task machine as unavailable for running jobs. The task machine will not be listed among task machines available to the program when you issue a BATCH QUERY TASK command. The program is continuing to operate.

User Response: Take one of these actions:

1. If the task machine should not be available to the VM Batch Facility, edit the CONTROL FILE. Delete the TASK entry that defines the task machine named in the message. File the CONTROL FILE and issue the BATCH CONTROL SET command to put the change into effect.

2. If the task machine should be defined in the system, add a CP directory entry to define the virtual machine. Then issue the BATCH CONTROL SET command, or stop and restart the VM Batch Facility, to make the task machine available to the program.

DGRVMC006W Error linking to task *userid* **191 minidisk; CP LINK rc =** *rc*

Explanation: The VM Batch Facility was unable to link to the 191 minidisk of the task machine identified in the message. The CP LINK command failed with the return code specified in the message.

The program links to the 191 minidisk of a task machine before a job is started and when a job is ended. The purpose of the link is to erase all job files from the 191 minidisk to help ensure job security.

VM Batch Facility Action: The VM Batch Facility is continuing to operate. However, the program will not be able to use the task machine identified in the message until the LINK problem is corrected. The task machine has been marked OFFLINE and will not be used until:

• It is restarted with the BATCH START command; or

• The VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is issued. These two events automatically reset an OFFLINE task machine.

User Response: Look up the return code in the message in the description of the LINK command. The command is described in your VM system library.

When the problem has been corrected, use the BATCH START command to restart the task machine. The program should resume normal use of the task machine.

DGRVMC007I Task machine *userid* **has been forced**

Explanation: The VM Batch Facility has logged off the task machine identified in the message. The message is sent only to the monitor machine's console for recording in the console message log file, if console logging is active.

The task machine has been forced because the VM Batch Facility was unable to carry out a directory-update-in-place to change the account number to be used by the task machine.

VM Batch Facility Action: The VM Batch Facility is continuing to operate, but cannot use the task machine shown in the message.

User Response: None is required.

DGRVMC008W The VM Batch Facility is trying to start job *jobid* **on your user ID.**
            Please log off or change the start time of the job

Explanation: The VM Batch Facility is trying to start a job that you submitted to run under the AUTOLOG job class. AUTOLOG jobs are run on the virtual machine of the person who submitted the job. The job can be started only if you are not logged on to your user ID (virtual machine).

VM Batch Facility Action: The VM Batch Facility will try to start the job again once a minute and will send you this message every 5 minutes until the job is started, changed, or canceled. The program will not automatically cancel the job.

User Response: If you would like to have the class AUTOLOG job run at this time, you should log off so that the job can be started.

You can also take one of these actions:

- Use the BATCH CHJOB command or the Chjob panel to set a new start window open time for the job. Specify a Start Window Open or WBEGIN time after the time you plan to log off. The VM Batch Facility will hold the AUTOLOG job until then.

- Cancel the job and submit it again later.

DGRVMC009E Job *userid nodeid jobid* **canceled: invalid logon password provided**

Explanation: The VM Batch Facility was unable to start the CLASS AUTOLOG job named in the message. The job could not be started because the logon password associated with the job was incorrect. The program could not log on your virtual machine to run the job.

VM Batch Facility Action: The job named in the message has been canceled.

User Response: Submit the job again. Be sure that you specify the correct logon password for the user ID from which you are submitting the job. Use the PASSWORD option or panel field to specify the password.

If you have submitted other CLASS AUTOLOG jobs that have not run yet, you can update the passwords of those jobs with the BATCH PASSWORD command or Password panel, or with the BATCH CHJOB command or Chjob panel.

DGRVMC011E Job *userid nodeid jobid* **canceled: error on AUTOLOG command, rc =** *rc*

Explanation: The VM Batch Facility was unable to start the CLASS AUTOLOG job named in the message. The AUTOLOG command that the program issued failed with the return code shown in the message. The message is sent to the monitor machine's console for recording in the console message log file.

VM Batch Facility Action: The job named in the message has been canceled.

User Response: Look up the return code shown in the message in the description of the AUTOLOG command. The command is described in your VM system library. Correct the problem and have the job owner submit the job again.

DGRVMC012E Invalid storage for task machine *userid*, **class** *classid*

Explanation: The VM Batch Facility was unable to AUTOLOG the task machine named in the message because of a problem with virtual storage. The problem is probably that the job class named in the message requires more virtual storage than the task machine is allowed to have.

VM Batch Facility Action: The VM Batch Facility has marked the task machine as OFFLINE. The program will not try to use the task machine again until:

- The BATCH START command has been used to restart the task machine; or

- The VM Batch Facility has been stopped and restarted or a BATCH CONTROL SET command has been issued. These events automatically reset an OFFLINE task machine.

User Response: Use the BATCH QUERY CLASS *classid* command to review the definition of the job class named in the message. Remember that jobs retain the virtual storage requirements that were defined for their job classes when they were received. A subsequent reduction in the virtual storage defined for a job class will not reduce the virtual storage required by jobs that are already queued in the job class.

Then, compare the virtual storage required by the job class to the maximum virtual storage allowed to the task machine. You can check on the task machine's virtual storage by reviewing the CP directory entry of the task machine.

If the job class storage is higher than the task virtual maximum storage, you should either:

- Increase the maximum virtual storage allowed to the task machine. The maximum virtual storage must be at least as large as the storage required by the job class.

- Decrease the virtual storage defined for the job class. The job class requirement must not exceed the maximum virtual storage available to the task virtual machine.

  Note: Consider canceling or changing the job class of all jobs already queued in the job class. Altering the job class definition will not change the virtual storage required for jobs that are already queued in the job class. A queued job will be started with the virtual storage that was defined for the job class when the job was received. The problem will continue to occur until the job is canceled or the task machine's storage is increased.

When the correction has been made, use the BATCH START *userid* command to make the task machine available to the VM Batch Facility again.

DGRVMC013E Job *userid nodeid jobid* **canceled: invalid account number**

Explanation: The account number specified for the job named in the message was invalid. The VM Batch Facility has canceled the job.

VM Batch Facility Action: The VM Batch Facility has canceled the job named in the message.

User Response: Check to be sure that you used an account number that you are permitted to use and that you have entered the number correctly. Then submit the job again.

DGRVMC014E Directory update for task *userid* **failed, DIAGNOSE X'84' rc =** *rc*

Explanation: The VM Batch Facility was unable to perform a directory-update-in-place of the task machine named in the message. The DIAGNOSE X'84' instruction failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility was not able to use the task machine named in the message to run a job. The program has marked the task machine as OFFLINE and will not try to use it again until:

- It is made available again with the BATCH START command.

- The VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is issued. These events automatically reset OFFLINE task machines.

The job that was being started remains queued. It will be started when a task machine that can accept it becomes available.

User Response: Look up the return code shown in the message in the description of the DIAGNOSE X'84' instruction. The instruction is described in your VM system library. Correct the problem indicated by the return code. Make the task machine available to the VM Batch Facility again by issuing the BATCH START command.

DGRVMC015W Error accessing task *userid* **191 disk, CMS ACCESS rc =** *rc*

Explanation: The monitor virtual machine could not access the 191 minidisk of the task machine named in the message. The ACCESS command issued by the monitor machine failed with the return code shown in the message.

The failure occurred after a job had ended or before a new job is started, when the VM Batch Facility was attempting to erase the task machine 191 minidisk. The error did not cause any job to be canceled; if a job was being started when the error occurred, the job remain queued.

VM Batch Facility Action: The VM Batch Facility has marked the task machine as OFFLINE. The program will not try to use the task machine again until:

- The BATCH START command has been used to restart the task machine.

- The VM Batch Facility has been stopped and restarted or a BATCH CONTROL SET command has been issued. These events automatically reset OFFLINE task machines.

User Response: Refer to the description of the ACCESS command in your VM system library. Correct the problem indicated by the return code. Enter the BATCH START *userid* command to make the task machine available to the VM Batch Facility again.

DGRVMC016W Error writing to task *userid* **191 disk, FSWRITE rc =** *rc*

Explanation: The monitor virtual machine could not write to the 191 minidisk of the task machine named in the message. The FSWRITE macro issued by the monitor machine failed with the return code shown in the message.

The failure occurred after a job had ended, when the VM Batch Facility was attempting to erase the job files from the task machine minidisk. The error did not cause any job to be canceled.

VM Batch Facility Action: The VM Batch Facility has marked the task machine as OFFLINE. The program will not try to use the task machine again until:

- The BATCH START command has been used to restart the task machine.

- The VM Batch Facility has been stopped and restarted or a BATCH CONTROL SET command has been issued. Both of these events automatically reset OFFLINE task machines.

User Response: Refer to the description of the FSWRITE macro in your VM system library. Correct the problem indicated by the return code. Enter the BATCH START *userid* command to make the task machine available to the VM Batch Facility again.

DGRVMC017E Job *userid nodeid jobid* **canceled, PUNCH rc =** *rc*

Explanation: The monitor virtual machine was unable to send the job named in the message to a task machine. The PUNCH command issued by the monitor machine failed with the return code shown in the message.

VM Batch Facility Action: The job has been canceled and will not be run. The VM Batch Facility is continuing to operate.

User Response: Refer to the description of the PUNCH command in your VM system library. Correct the problem indicated by the return code and have the job owner submit the job again.

DGRVMC018E Could not spool punch to task machine *userid*, **CP SPOOL rc =** *rc*

Explanation: The monitor virtual machine was unable to spool its virtual punch to the task machine named in the message. The SPOOL command issued by the monitor machine failed with the return code shown in the message.

The monitor machine sends job files to a task machine through its virtual punch. Before it can PUNCH a file to a task machine, it must spool the punch to the task machine.

VM Batch Facility Action: The task machine named in the message will be put into OFFLINE status. The VM Batch Facility will not use the task machine until:

• It is released from OFFLINE status with a BATCH START command.

• The VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is issued. Both of these events automatically reset OFFLINE task machines.

User Response: Refer to the description of the CP SPOOL command in your VM system library. Correct the problem indicated by the return code shown in the message.

Then enter the BATCH START *userid* to make the task machine available for running jobs again.

DGRVMC019E Job *userid nodeid jobid* **canceled, CP TRANSFER rc =** *rc*

Explanation: The monitor virtual machine was unable to send the job file named in the message to a task machine. The TRANSFER command issued by the monitor machine failed with the return code shown in the message.

The VM Batch Facility uses the TRANSFER command to send a job file when the job file has been queued in the monitor machine punch. Jobs are queued in the punch if they are too large to fit on the monitor machine's 193 minidisk, where jobs files are usually stored.

VM Batch Facility Action: The job named in the message has been canceled and will not be run. The VM Batch Facility is continuing to operate.

User Response: Refer to the description of the TRANSFER command in your VM system library. Correct the problem indicated by the return code and have the job owner submit the job again.

DGRVMC020E Error setting task *taskid* **as alternate user for** *userid*, **DIAGNOSE X'D4'**
        rc = *rc*

Explanation: The monitor machine was unable to set up a task machine as an alternate user ID for a job owner. The user IDs of the task machine and job owner appear in the message. The DIAGNOSE X'D4' issued by the monitor machine failed with the return code shown in the message.

VM Batch Facility Action: The VM Batch Facility marks the task machine as OFFLINE and cancels the job that was being started. The task machine will not be used again until:

• It has been made available again with the BATCH START command.

• The VM Batch Facility has been stopped and restarted or a BATCH CONTROL SET command has been issued. Both of these events automatically reset OFFLINE task machines.

The job must be submitted again in order to be run.

User Response: Refer to the description of DIAGNOSE X'D4' in your VM library. Correct the problem indicated by the return code shown in the message.

DGRVMC021E Autolog for task *userid* **failed, AUTOLOG rc =** *rc*

Explanation: The VM Batch Facility was unable to start a job on the task machine named in the message. The AUTOLOG command that was issued to log on the task machine failed with the return code shown in the message. The message is sent to the monitor machine console for recording in the console message log, if console message spooling is active.

VM Batch Facility Action: The VM Batch Facility has marked the task machine named in the message as OFFLINE. The program will not try to use the task machine again until:

- The BATCH START command is issued to make the task machine available again.

- The VM Batch Facility is stopped and restarted or a BATCH CONTROL SET command is issued. Both of these events automatically reset OFFLINE task machines.

User Response: Refer to the description of the AUTOLOG command in your VM system library. Correct the problem indicated by the return code and make the task machine available to the VM Batch Facility again.

DGRVMC5nn

User Response: See "Messages Numbered 500 and Higher."

Messages Numbered 500 and Higher

The messages in this section are common to more than one VM Batch Facility component, and are identified by a unique number of 500 or greater. Look up the message using the message number – the seventh through ninth characters of the message ID.

500I   New LLS subject jobs are held because of resource contention

Explanation: The Suspend Contention Value for at least one resource has been exceeded. The Load Level Scheduler will prevent all queued jobs that are subject to LLS from being dispatched.

VM Batch Facility Action: Queued LLS subject jobs will not be dispatched until contention for system resources decreases below the Resume Contention Values, and all suspended jobs have resumed.

User Response: None required.

501I   Job *userid nodeid jobid* **on task** *taskid* **suspended for contention of** *resource***, MCV =** *mmm***, SCV =** *sss*

Explanation: The job has been suspended by the Load Level Scheduler, due to the level of contention for *resource*. *mmm* is the Measured Contention Value for the resource, *sss* is the defined Suspend Contention Value.

VM Batch Facility Action: The job will be resumed when contention for the resource decreases below the Resume Contention Value.

User Response: None required.

502I   New LLS subject jobs will be dispatched

Explanation: A state of contention has ended. LLS subject jobs in the queue will no longer be prevented from dispatch.

VM Batch Facility Action: LLS subject jobs will start, but will remain subject to suspension should contention for system resources arise.

User Response: None required.

503I   **Job** *userid nodeid jobid* **on task** *taskid* **resumed after contention of** *resource*;
           MCV = *mmm*, **RCV =** *rrr*

Explanation: A job previously suspended by the Load Level Scheduler, due to high levels of contention for *resource*, has now been resumed. *mmm* is the Measured Contention Value, *rrr* is the Resume Contention Value for *resource*.

VM Batch Facility Action: The job resumes, but will remain subject to suspension should contention for system resources increase.

User Response: None required.

504W   **Thrash of Load Level Scheduling detected for** *resource*
           MCV = *mmm*, **SCV =** *sss*, **RCD =** *ddd*, **RCV reduced to** *rrr*

Explanation: The Load Level Scheduler is thrashing. The Suspend Contention Value (SCV) and the Resume Contention Value (RCV) for the resource are too close, resulting in jobs being continually suspended and resumed.

VM Batch Facility Action: The RCV is reduced by the defined RCD, in order to avoid further thrashing.

User Response: The value defined for the RCD in the Control File should be increased. This will cause the RCV to be set at a lower value the next time the VM Batch Facility is started.

505I   **Job** *userid nodeid jobid* **not subject to LLS**

Explanation: This message is issued in response to a BATCH RESUME command or a BATCH CHJOB command that specified the option NOLLS, indicating that the job is now no longer subject to Load Level Scheduling.

VM Batch Facility Action: This job is no longer a candidate for suspension by the LLS when system contention occurs.

User Response: None.

506I   **Job** *userid nodeid jobid* **subject to LLS**

Explanation: This message is issued in response to a BATCH CHJOB command that specified the option LLS; indicating that the job is now subject to Load Level Scheduling.

VM Batch Facility Action: This job is now a candidate for suspension by the LLS when system contention occurs.

User Response: None.

507E   **LLS values are invalid**

Explanation: This message is issued in response to a BATCH LLS *resource llsscv llsrcd* command, or to record an invalid specification of values for an LLS keyword in the Control File.

VM Batch Facility Action: If issued as a response to an LLS command, the command is ignored and Load Level Scheduler activity remains unchanged. If issued as a result of an invalid specification of LLS values in the Control File, the keyword is ignored and LLS will be inactive for the specified resource.

User Response: *resource* is invalid if not one of the following:

1. CPU
2. STO
3. PAG
4. IO

Both *llsscv* and *llsrcd* must be positive integers greater than zero, or an asterisk (*).

llsscv must be greater than *llsrcd*. Re-issue the command with valid values.

508I   **LLS is active for** *resource*
             MCV = *mmm*, **SCV =** *sss*, **RCD =** *ddd*, **RCV =** *rrr*
             contention status; *ccc* **running jobs are suspended**

Explanation: This message is issued for each active LLS resource in response to a
BATCH LLS command.

mmm is the Measured Contention Value for the resource, *sss* is the defined Suspend
Contention Value, *ddd* is the defined Resume Contention Decrement, *rrr* is the computed
Resume Contention Value.

contention status is a verbal summary of the contention values. It may be *No Contention
Exists*, or *Contention Exists*, or *Contention is Resolving*, or *No Contention Values*.

No contention exists when the MCV is less than the SCV and no jobs are suffering any LLS
degradation. Contention exists when the MCV exceeds the SCV and LLS subject jobs are
being progressively degraded. Contention is resolving when the MCV has fallen below the
RCV, and previously degraded jobs are being resumed; however the resumption is not yet
complete for all jobs. No measured contention values are available immediately after
monitor startup, or after a period of Facility inactivity, as these are averaged over a time
period before being accepted as reliable. In this case the monitor behaves as if contention
did not exist.

VM Batch Facility Action: None.

User Response: None.

509I   **LLS is not active for** *resource*

Explanation: This message is issued for each inactive LLS resource in response to a
BATCH LLS command.

VM Batch Facility Action: None.

User Response: None.

510I

Explanation: This message is issued to the VM Batch Facility console to record the receipt
of a BATCH LLS command.

VM Batch Facility Action: None.

User Response: None.

511E   **Job** *userid nodeid jobid* **rejected; LLS override is invalid**

Explanation: The job has been rejected by the VM Batch Facility because an override of
the default job subjection to LLS for the job's class was specified, and the Class definition
does not permit the requested override.

VM Batch Facility Action: The job is not accepted to the Facility.

User Response: Either remove the override request from the BATCH SUBMIT command,
or submit the job to a class that permits the required LLS subjection.

512E   **Chjob** *userid nodeid jobid* **rejected; LLS override is invalid**

Explanation: A CHJOB command has been rejected because an override of the default job
subjection to LLS for the job's class was specified, and the Class definition does not permit
the requested override.

VM Batch Facility Action: The job is not changed.

User Response: An unauthorized user may not override a job's LLS subjection if the job's
Class does not permit it.

An authorized administrator may unconditionally change a job's subjection to LLS activity by issuing the CHJOB command.

513E   Command *command* **is not available via** *exit* **exit**

Explanation: A command has been received by the monitor from an installation exit, but is either not recognized, or is not permitted from an exit.

VM Batch Facility Action: The command is ignored.

User Response: Correct the exit that issued the command.

514I   Job *userid nodeid jobid* **held**

Explanation: This message is issued in response to a BATCH HOLD command, indicating that the job is now prevented from being dispatched.

VM Batch Facility Action: The job will not be dispatched until a FREE command is issued for it.

User Response: None.

515I   Job *userid nodeid jobid* **freed**

Explanation: This message is issued in response to a BATCH FREE command, indicating that the job is now available for dispatch.

VM Batch Facility Action: The job will be dispatched when selected by the dispatcher.

User Response: None.

516I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH HOLD command.

VM Batch Facility Action: None.

User Response: None.

517I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH FREE command.

VM Batch Facility Action: None.

User Response: None.

518I   Job *userid nodeid jobid* **is queued**

Explanation: This message is issued to the monitor console to indicate that a submitted job has been accepted for processing by the Facility.

VM Batch Facility Action: The job will be dispatched when selected by the dispatcher.

User Response: None.

519I   ALTID *altuser* **set for** *jobuser jobnode jobname*

Explanation: An Alternate Userid has been confirmed successfully by an ALTID command.

VM Batch Facility Action: The job will be started with the specified Alternate Userid.

User Response: None required.

520I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a command from an exit.

VM Batch Facility Action: None.

User Response: None.

521W   Invalid syntax following keyword parameter *keyword*

Explanation: While reading the Control File, a statement that was coded using keyword parameters was syntactically invalid. *keyword* indicates the last valid keyword found in the statement.

This message is followed by another identifying the Control File statement.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

522W   Keyword parameter *keyword* **not recognized**

Explanation: While reading the Control File, a statement that was coded using keyword parameters specified a keyword that was not meaningful in the statement.

This message is followed by another identifying the Control File statement.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

523W   Keyword parameter *keyword* **missing, or value is invalid**

Explanation: While reading the Control File, a statement that was coded using keyword parameters either failed to specify a mandatory keyword, or a value assigned to a keyword was not valid.

This message is followed by another identifying the Control File statement.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

524W   Keyword parameter or value is too long

Explanation: While reading the Control File, a statement that was coded using keyword parameters contained a string that was too long to be a valid keyword or value. The Control File record containing the offending string follows.

This message is followed by another identifying the Control File statement.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

525W   Class definition ignored

Explanation: While reading the Control File, a Class definition statement that was coded using keyword parameters contained an error resulting in the definition being ignored. The FIRST line of the offending statement follows for the purpose of identifying the statement. It is not necessarily the line containing the error.

This message is preceded by another identifying the nature of the error.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

526I   **Job** *userid nodeid jobid* **is changed**

Explanation: This message is issued to the monitor console to indicate that a CHJOB command has been accepted and successfully processed by the Facility.

VM Batch Facility Action: The job will behave in accordance with its new job control values.

User Response: None.

527W   **LLS definition ignored**

Explanation: While reading the Control File, an LLS definition statement that was coded using keyword parameters contained an error resulting in the definition being ignored. The FIRST line of the offending statement follows for the purpose of identifying the statement. It is not necessarily the line containing the error.

This message is preceded by another identifying the nature of the error.

VM Batch Facility Action: The Control File statement is ignored.

User Response: Correct the statement in the Control File.

528E   **Exit** *exit* **issued more than five commands**

Explanation: An exit has issued more than five commands to the monitor after a single invocation of the exit. A maximum of five is permitted.

VM Batch Facility Action: All the commands issued by the exit for this invocation are ignored.

User Response: Correct the exit to obey the limitation of five commands per invocation.

529E   **Insufficient virtual storage; Exit commands ignored**

Explanation: There was insufficient virtual storage to process commands issued by an exit.

VM Batch Facility Action: All the commands issued by the exit for this invocation are ignored.

User Response: Increase the virtual storage available to the monitor virtual machine.

530I   **Job** *userid nodeid jobid* **will be resumed**

Explanation: This message is issued in response to a CHJOB command that changed a job from subject to independent of LLS when the job had already been suspended for contention.

VM Batch Facility Action: The job will be resumed when the monitor performs its check of LLS activity.

User Response: None.

531E   **An option or keyword is invalid; Command from** *exit* **ignored**

Explanation: A command issued by an exit contained an invalid value for an option or keyword. The command in error follows the message.

VM Batch Facility Action: The command is ignored.

User Response: Correct the exit.

532I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH SUBMIT command.

VM Batch Facility Action: None.

User Response: None.

533I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH CANCEL command.

VM Batch Facility Action: None.

User Response: None.

534E   **Errors while processing Exit Commands; Check Monitor console**

Explanation: This message is issued to the VM Batch Facility alertee when errors have been detected in the processing of commands issued by an exit.

VM Batch Facility Action: Each error is logged to the monitor console.

User Response: Correct the errors in the exit.

535E   **Job** *userid nodeid jobid* **not queued; Command** *command* **ignored**

Explanation: A command has been issued against a job that requires that the job be queued (as opposed to running or completed) for successful execution of the command.

VM Batch Facility Action: The command is ignored.

User Response: Ensure that the correct command is being issued.

536W   **Altid** *altid* **for job** *userid nodeid jobid* **not confirmed**

Explanation: A job that requires confirmation of the alternate user ID under which it is to run is about to start, but no ALTID command has been received to confirm it.

VM Batch Facility Action: No alternate userid is set for the job by the monitor; but the DGRIDO exit is called for reason RJEALTU and it may directly set an alternate userid for the job.

User Response: Ensure that the DGRUJB exit is confirming alternate userid's when required.

538I   ***END OF QUERY**

Explanation: This message terminates a group of messages that comprise the response to a BATCH QUERY command.

VM Batch Facility Action: None.

User Response: None.

539E   **Job is ADMIN-FREED; cannot be held by an exit**

Explanation: A BATCH HOLD command has been issued indirectly or by return code from an exit, but is rejected as the job was previously freed by an administrator.

VM Batch Facility Action: The job remains free.

User Response: This message should be expected when an administrator frees a job to override an automatic HOLD from an exit.

540E   Job is ADMIN-HELD; cannot be freed by an exit

Explanation: A BATCH FREE command has been issued indirectly or by return code from an exit, but is rejected as the job was previously held by an administrator.

VM Batch Facility Action: The job remains held.

User Response: This message should be expected when an administrator holds a job to override an automatic FREE from an exit.

541I   Job *userid nodeid jobid* **canceled. [Job will recur]**

Explanation: This message confirms the cancellation of a job. The job will recur if the canceled job was a recurring job instance and the job was not FLUSHed.

VM Batch Facility Action: Cancellation of the job has been initiated. If the job was not running, then this message also means that the cancellation is complete. For running jobs, cancellation is complete only when the processing specified for the ERROR option of the job has also completed; however, ERROR processing is not performed after a BATCH STOP JOBS command.

User Response: If no accompanying message explains the reason for the cancellation, then issue the BATCH STATUS command for more detail.

542W   Job *userid nodeid jobid* **canceled by exit** *exit* **[Job will recur.]**

Explanation: This message confirms the cancellation of a job, when the cause of cancellation is a specific (as opposed to default) request by an exit. The job will recur if the canceled job was a recurring job instance and the job was not FLUSHed.

VM Batch Facility Action: The job has been canceled.

User Response: Contact your VM Batch administrator to find out why the exit canceled the job.

543E   Spool file purged, possible security violation

Explanation: A spool file originating from a remote node has been detected with the transfer bit, in its spool file block, set. The VM Batch Facility is unable to verify the contents of the tag information in this case.

VM Batch Facility Action: The spool file is purged.

User Response: This condition may result in incorrect information being obtained by the VM Batch Facility, and therefore the job is canceled, and follow-up action is left to the discretion of the VM Batch Facility system administrator. Additional information on the file may be obtained from preceding messages in the monitor machine's console log, and from the RCCS virtual machine.

544E   Command *command* **is not permitted within a job**

Explanation: A VM Batch Facility command has been issued from within a VM Batch Facility job, but is not permitted in this environment.

VM Batch Facility Action: The command is ignored. A return code of 12 is passed to the invoking job EXEC.

User Response: Correct the job that issued the command.

545E   **Usage of** *command* **is incorrect from within a job**

Explanation: A VM Batch Facility command that is permitted from within a VM Batch Facility job has been issued, but does not conform to the restrictions imposed on the command when issued from within a job.

VM Batch Facility Action: The command is ignored. A return code of 8 is passed to the invoking job EXEC.

User Response: Correct the job that issued the command.

546E   **Job/Chjob** *userid nodeid jobid* **rejected; The valid storage range for class** *classid* **is** *minstor* **to** *maxstor*

Explanation: The job or command has been rejected by the VM Batch Facility because an override of the default virtual storage allocation for the job's class was specified, and the class definition does not permit the requested storage value.

VM Batch Facility Action: The job or command is rejected.

User Response: Ensure that the storage value is within the bounds available to the job class, or submit the job under a different class with adequate storage bounds.

547E   **Too many invalid AUTOLOG passwords**

Explanation: CP has disabled the AUTOLOG command for the issuing VM Batch Facility monitor, because too many invalid passwords have been encountered on CP AUTOLOG commands when starting jobs.

VM Batch Facility Action: The monitor terminates when all running non class AUTOLOG jobs have ended.

User Response: The monitor virtual machine must be logged off and on again in order to restore the CP AUTOLOG capability.

You may wish to terminate the monitor immediately, without waiting for running jobs to complete, by issuing the BATCH STOP NOW command.

If your system does not require passwords for CP AUTOLOG commands, you may configure the VMBATCH DEFAULTS file to prevent accidental use of inappropriate passwords.

548I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH RELOAD command.

VM Batch Facility Action: None.

User Response: None.

549I   **Installation exits have been reloaded**

Explanation: This message is issued in response to a BATCH RELOAD command, indicating that the command has been processed successfully.

VM Batch Facility Action: The monitor is running with the new installation monitor exits.

User Response: None.

550E   **Task machine** *taskid* **has been put offline**

Explanation: An event has occurred that has made the identified task machine unusable.

VM Batch Facility Action: No further jobs are sent to the task machine for execution.

User Response: Examine the monitor's console to determine why the task machine has been put offline. The reason is stated in a message preceding this one on the console.

Once the problem has been corrected, the task machine may be brought back into service by issuing the BATCH START command, or by issuing the BATCH CONTROL SET command, or by stopping and restarting the monitor.

551I   **AUTOLOG passwords are discarded**
551E   **AUTOLOG passwords are discarded**

Explanation: The VMBATCH DEFAULTS file contains the statement AUTOPASS = NO, requesting that the VM Batch Facility monitor not use any supplied passwords whenever a CP AUTOLOG command is issued.

VM Batch Facility Action: At monitor initialization time, this is an informative message only; monitor processing continues. On receipt of a class AUTOLOG job that has supplied a password, the password has been discarded but the job is accepted. In response to a BATCH PASSWORD command, the command is rejected.

User Response: Submit all class AUTOLOG jobs in accordance with your installation's requirements for passwords. Do not use the BATCH PASSWORD command if it is not appropriate at your installation.

552I   **AUTOLOG passwords are required**
552E   **AUTOLOG passwords are required**

Explanation: The VMBATCH DEFAULTS file contains the statement AUTOPASS = YES, requesting that the VM Batch Facility monitor insist on passwords being supplied whenever a CP AUTOLOG command is to be issued.

VM Batch Facility Action: At monitor initialization time, this is an informative message only; monitor processing continues.

When issued in response to a BATCH PASSWORD command, it means that a password of *NOPASS* was specified on the command. This is not acceptable to the monitor, so the command is rejected.

User Response: If this message was received in response to a BATCH PASSWORD command, then notify your VM Batch Facility administrator that your VMBATCH DEFAULTS file does not match that of the monitor you are addressing.

553W   **Invalid VMBATCH DEFAULTS file record ignored**

Explanation: The VMBATCH DEFAULTS file contains an invalid statement. The invalid statement follows this message.

VM Batch Facility Action: The record is ignored.

User Response: Have a VM Batch Facility administrator correct the VMBATCH DEFAULTS file.

554E   **Job** *userid nodeid jobid* **rejected; no logon password was provided**

Explanation: The Facility has been configured to insist that logon passwords are provided for class AUTOLOG jobs, and no logon password has been supplied for this class AUTOLOG job.

VM Batch Facility Action: The job is rejected.

User Response: Resubmit the job with a valid password.

555I   *Modname* **Version** *ver.rel***, Service Level** *nnn*

Explanation: This is the version, release, and service level associated with the executable component *modname*.

VM Batch Facility Action: None required.

User Response: None.

556W   LOCJCL card truncated

Explanation: The DGREVR user interface installation exit has provided an installation defined job control (LOCJCL) card to be submitted with your job, and the card text is too long.

VM Batch Facility Action: The job is submitted, but the text of the offending LOCJCL card has been truncated to 73 characters. A return code of 4 is passed to the invoker.

User Response: A VM Batch Facility administrator must correct the user interface installation exit to comply with the documented requirements for LOCJCL cards.

557W   Extraneous LOCJCL cards were discarded

Explanation: The DGREVR user interface installation exit has provided more than three installation defined job control (LOCJCL) cards to be submitted with your job.

VM Batch Facility Action: The job is submitted, but only three LOCJCL cards have been included in the job deck. The others have been discarded. A return code of 4 is passed to the invoker.

User Response: A VM Batch Facility administrator must correct the user interface installation exit to comply with the documented requirements for LOCJCL cards.

558E   Job/Chjob *userid nodeid jobid* **rejected; recurrence pattern** *recurrence* **not accepted**

Explanation: The SUBMIT or CHJOB command has been rejected by the VM Batch Facility because the value specified for the REPEAT option was not accepted by the recurrence installation exit, DGRECUR EXEC.

VM Batch Facility Action: The job or command is rejected.

User Response: Ensure that the value supplied for the REPEAT option is available at your installation. Issue BATCH QUERY REPEATS to learn what recurrence options are available at your installation, or consult your VM Batch Facility administrator.

559I   Job *userid nodeid jobid* **flushed; job will not recur**

Explanation: A recurring job has been canceled and the FLUSH option was specified on the CANCEL command. Any executing instance of the job has been canceled, and the job was not re-queued to run again.

VM Batch Facility Action: The job has been discarded.

User Response: None.

560E   Chjob *userid nodeid jobid* **rejected; REPEAT option cannot be specified for chained jobs**

Explanation: The CHJOB command has been rejected by the VM Batch Facility because a REPEAT option was specified on the command, and the job is chained to another. Chained jobs acquire an implicit recurrence when chained to a recurring job, but may not have an explicit recurrence set.

VM Batch Facility Action: The command is rejected.

User Response: Re-issue the command without specifying the REPEAT option. Alternatively, if an explicit recurrence is required for the job, then cancel the job and re-submit it with the REPEAT option.

561W   **Recurring job** *userid nodeid jobid* **has not been rescheduled; job is held**

Explanation: The recurrence exit DGRECUR EXEC has not queued any command in response to a call from the monitor requesting rescheduling of the job.

VM Batch Facility Action: The job is re-queued with DATE HOLD.

User Response: Ensure that the recurrence exit is behaving properly in that it should respond to a call with action = 'NEXT' with an internal CHJOB command to set the date and time of the next occurrence of the job. You may set the next occurrence manually by issuing a CHJOB command directed at the job, specifying the DATE and WBEGIN options.

562I   **Job** *userid nodeid jobid* **selected for dispatch**

Explanation: All scheduling criteria have been satisfied for the job. A suitable task machine is available, and the job will be started immediately subject only to confirmation by the relevant installation exits.

VM Batch Facility Action: The job starts immediately unless prevented by an installation exit.

User Response: None.

563E   **Chjob** *userid nodeid jobid* **rejected; CHAIN option cannot be used on explicitly recurring jobs**

Explanation: The CHJOB command has been rejected by the VM Batch Facility because a CHAIN option was specified in the command and the job is explicitly recurring.

VM Batch Facility Action: The command is rejected.

User Response: Re-issue the command without specifying a CHAIN option. Alternatively, if chaining is required for the job, then issue a CHJOB command with REPEAT NEVER to negate its explicit recurrence, and then chain it.

564E   **Job/Chjob** *userid nodeid jobid* **rejected; insufficient space on the** *monitor* **193 minidisk**

Explanation: A recurring job has been submitted and the job has been rejected by the VM Batch Facility because there is no room for the monitor to store the job on its 193 disk. Recurring jobs must be stored on disk for reasons of system integrity.

VM Batch Facility Action: The job is rejected.

User Response: Approach your Facility administrator to provide sufficient disk storage to allow your recurring job to be saved. The job may of course be submitted in the meantime without recurrence, that is, with a REPEAT value of 'NEVER'.

565E   **Temporary file failure;** *reason*

Explanation: The VM Batch Facility SUBMIT XEDIT macro has not submitted the job to the VM Batch Facility because the temporary file needed to hold the job during the submission process could not be created; *reason* indicates the nature of the difficulty.

VM Batch Facility Action: The job is not submitted. The SUBMIT XEDIT macro returns a code of 16.

User Response: Correct the situation indicated by *reason* and then re-issue the SUBMIT XEDIT macro.

566W   **Job** *userid nodeid jobid* **held; no task machine can run the job**

Explanation: The job has been held by the VM Batch Facility because no task machine can run the job. This may be for one of the following reasons:

1. There are no task machines that can run jobs of the specified class.

2. There are no task machines with a window that is compatible with the job start window that was specified by the WBEGIN and WEND options of the SUBMIT or CHJOB command.

VM Batch Facility Action: The job is held.

User Response: If you did not intend the job to be held, issue a CHJOB command to alter the job's class and/or start window so that it is compatible with a suitable task machine. The class and window settings for the task machines may be obtained using the BATCH QUERY TASK command.

This message may also arise when a VM Batch Facility has been reconfigured and an existing viable job now finds itself without a suitable task machine. In this case you should discuss any possible problems with your VM Batch Facility administrator.

567W   **Job** *userid nodeid jobid* **flushed by chain**

Explanation: The identified job was chained to a recurring job, and that parent job was flushed; so it has also been flushed.

VM Batch Facility Action: The job has been discarded.

User Response: None.

568I   **Number of jobs left running:** *jobcount*

Explanation: The monitor has shutdown and *jobcount* jobs were left running.

This message is issued when the monitor was stopped by a STOP NOW command, and at least one job was left running.

VM Batch Facility Action: None.

User Response: None.

569I   **Number of jobs canceled:** *jobcount*

Explanation: The monitor has shutdown and *jobcount* jobs were canceled.

This message is issued when the monitor was stopped by a STOP JOBS command, and at least one job was cancelled.

VM Batch Facility Action: None.

User Response: None.

570I   **Executing process** *proc* **in** *csect* **for** *subject*

Explanation: This message identifies an internal process executed by a VM Batch Facility component.

VM Batch Facility Action: None.

User Response: None.

571E   **Please set MSG ON before using BATCH LEVEL**

Explanation: In response to a BATCH LEVEL command, the user's virtual machine MSG setting was found to be OFF. For the results of the BATCH LEVEL command to be returned to the user's virtual console, the CP SET MSG ON command must be executed.

VM Batch Facility Action: The command terminates with return code 16.

User Response: Issue the CP SET MSG ON command before executing BATCH LEVEL.

572W   **A password is required. Enter the required password or press Enter to cancel**

Explanation: The VM Batch Facility has been configured to insist on a password being supplied when a class AUTOLOG job is submitted, and a null response was received in reply to the prompt for the password.

VM Batch Facility Action: The Facility prompts for the password a second time; if the response is again null then the job is not submitted.

User Response: Enter the password to have the job submitted, or just hit the Enter key to cancel.

573W   **Occurrence of job** *userid nodeid jobid* **prevented by exit DGRECUR**

Explanation: An instance of a recurring job was selected for dispatch, but a call to the recurrence exit DGRECUR EXEC resulted in a return code requesting that the job not run at this time. The exit should issue a command to schedule the job's next occurrence.

VM Batch Facility Action: The job is not run. It is re-queued with DATE HOLD, unless rescheduled otherwise by the exit.

User Response: Ensure that the recurrence exit is behaving properly in that having prevented the job from running at its scheduled time, it has also issued an internal CHJOB command to schedule the date and time of its next occurrence. You may set the next occurrence manually by issuing a CHJOB command directed at the job, specifying the DATE and WBEGIN options.

574E   **Job** *userid nodeid jobid* **not canceled; job will recur**

Explanation: This message is issued in response to a CANCEL command targeting a recurring job, and no FLUSH option was specified. No instance of the job was running, so the CANCEL command has had no effect.

VM Batch Facility Action: The job will be dispatched at its scheduled time.

User Response: None. If it was your desire to cancel the recurring job in a way that it would not be rescheduled, then re-issue the CANCEL command and specify the FLUSH option.

575E   **Command** *command* **from** *exit* **discarded; monitor is stopping**

Explanation: This message is issued to a monitor console when the monitor has been stopped by a STOP JOBS command, and an installation exit that was invoked because of a job being canceled has queued a command other than CHJOB to the monitor.

VM Batch Facility Action: The command is ignored.

User Response: You must decide if any of the discarded commands need to be re-issued manually when the monitor is restarted.

576E   **Synchronous submission did not complete**

Explanation: A job has been submitted with the WAIT option, and the submission did not complete.

This may be because the submission timed out, or was stopped by a terminal interrupt, or there was a transmission failure between the submitting virtual machine and the monitor.

VM Batch Facility Action: The submission is abandoned. The job will not run.

User Response: If this was not the result of an intentional termination of the submission, then it is most likely due to a submission timeout. If possible, correct the condition that has prevented timely delivery of the job. For example, if it was caused by the submission of a job to a VM Batch Facility monitor that is not running, then arrange for the monitor to be started before trying again.

Alternatively, if the cause of the timeout is beyond your control, for example when there are excessive network delays, you may consider increasing the value of the timeout that you have specified in the WAIT option.

The VM Batch Facility will attempt to delete the job. If the deletion is not successful, the job will lie dormant in the job queue without ever running. The response to a STATUS command on such a job would be 'Awaiting synchronization'. You should delete the job by a CANCEL command, since its presence in the queue will prevent the acceptance of a subsequent job with the same job id.

577I   **Job** *userid nodeid jobid* **awaiting synchronization**

Explanation: A job has been received that was submitted with the WAIT option specified on the SUBMIT command.

VM Batch Facility Action: The job is accepted, but will not be eligible to run until the synchronization of its submission has been confirmed. Message 578I or 579I should follow to indicate the success of the synchronization.

User Response: None.

578I   **Job** *userid nodeid jobid* **has synchronized**

Explanation: A job was submitted with the WAIT option specified on the SUBMIT command, and has completed its synchronized submission.

VM Batch Facility Action: The job is eligible to run according to the normal rules for job submission.

User Response: None.

579I   **Job** *userid nodeid jobid* **has been discarded**

Explanation: A job was submitted with the WAIT option specified on the SUBMIT command, and the submission was terminated before synchronization could complete.

This may be because the submission timed out, or was intentionally terminated by the submittor, or there were transmission problems between the submittor and the monitor.

This message may appear at monitor startup, once for each job that was synchronously submitted to the monitor while the monitor was down.

VM Batch Facility Action: The job is discarded.

User Response: None.

580E   **Job** *userid nodeid jobid* **rejected; no user block space available**

Explanation: A job was submitted and the target VM Batch Facility monitor did not have a control block available to allocate to the job.

VM Batch Facility Action: The job is rejected.

User Response: Notify your Facility administrator, who is able to allocate more resources to the monitor. In the meantime the situation may only be temporary; resubmit the job when the load on the facility has eased.

Administrators should note that the number of user blocks is controlled by the parameter MAXIMUM-USERS in the Control File.

581E   **Job** *userid nodeid jobid* **rejected; no job block space available**

Explanation: A job was submitted and the target VM Batch Facility monitor did not have a control block available to allocate to the job.

VM Batch Facility Action: The job is rejected.

User Response: Notify your Facility administrator, who is able to allocate more resources to the monitor. In the meantime the situation may only be temporary; resubmit the job when the load on the facility has eased.

Administrators should note that the number of user blocks is controlled by the parameter MAXIMUM-JOBS in the Control File.

582E   **Task** *taskid* **unable to start.**

Explanation: The task machine encountered one of the following problems:

- The VM Batch Facility task or user interface modules (DGRTSK MODULE, DGRTASK EXEC, DGRUVM MODULE or DGRRUN EXEC) were not available on any accessed disk.

- The VM Batch Facility message repository was not available on any accessed disk.

- There was insufficient virtual storage available to the task machine.

- The task machine was not able to issue CP DIAG x'68'.

VM Batch Facility Action: The task machine is marked offline.

User Response: Check the task machine environment for any of the items mentioned above.

583E   **Spawned command rejected; target job** *userid nodeid jobid* **was not spawned**

Explanation: A VM Batch Facility command that is permitted from within a VM Batch Facility job has been issued, but its target is a job that was not spawned. It is a restriction that commands issued from within a job may only act on jobs that were submitted from within a job.

VM Batch Facility Action: The command is ignored.

User Response: Correct the job that issued the command.

584E   **Command verb** *DGRINTxx* **reserved for IBM use**

Explanation: You have issued a VM Batch Facility command that is reserved for internal product use. Such commands are distinguished by a verb that begins with the string 'DGRINT'.

VM Batch Facility Action: The command is not transmitted to a VM Batch Facility monitor.

User Response: Ensure that any installation defined command verbs do not conflict with VM Batch Facility internal command verbs.

585E   **Invalid use of command** *command*

Explanation: The VM Batch Facility has detected an inappropriate use of a valid VM Batch Facility command. This situation may arise if a VM Batch Facility job attempts to issue commands other than those permitted to a task machine, a JOBSTEP command is issued by a normal user or required parameters were missing from a command issued via the RJE interface.

VM Batch Facility Action: The VM Batch Facility has ignored the command and is continuing to operate.

User Response: Make sure that commands are issued with the correct syntax and from the correct environment.

586E   **VM Batch Facility requires a 370-mode virtual machine**

Explanation: The VM Batch Facility monitor was started in an XA, ESA or XC mode virtual machine. A 370 mode virtual machine is required.

VM Batch Facility Action: The VM Batch Facility monitor does not start.

User Response: This is an obsolete message. You need to apply the latest VM Batch Facility service to run in XA, XC or ESA mode.

587E   **VM Batch Facility message repository not found**

Explanation: The VM Batch Facility requires that its message repository (filename DGRUME) be available.

VM Batch Facility Action: If issued as a result of a VM Batch Facility command, then the command has not been executed. If issued in response to an attempted startup of the VM Batch Facility, then the Facility has not started.

User Response: Contact your VM Batch Facility administrator to have the VM Batch Facility message repository made available.

588E   **VM Batch Facility component** *modname* **not found**

Explanation: In order to perform a required function, a necessary VM Batch Facility component *modname* was not available and the function failed.

VM Batch Facility Action: The VM Batch Facility command was not performed.

This message may also be issued by the Facility at monitor startup, in which case the monitor has failed to start.

User Response: Ensure that the required component is available on an accessed disk and re-issue the command.

589W   **This is the VM Batch Facility task machine interface**

Explanation: This message is issued by the VM Batch Facility task machine interface program when that program is invoked on a non-disconnected virtual machine.

VM Batch Facility Action: The task machine interface has terminated without performing any function.

User Response: Do not invoke the VM Batch Facility task machine interface in a virtual machine that is not a VM Batch Facility task machine. Insistence may result in the erasure all files on the 191 minidisk of the virtual machine on which it is running. If this message appears on the console of a bona fide task machine, then for the task machine to operate properly, it must run disconnected and not have any secondary consoles defined for it in the CP directory.

590I

Explanation: This is an internal product synchronization message. It may appear after a synchronous submission of a job with the WAIT option has timed out.

VM Batch Facility Action: None.

User Response: Please ignore the message.

591I **Monitor message code** *mcode* **accompanied return code** *rc* **on the synchronous submission of job** *userid nodeid jobid*

Explanation: The synchronous submission of a job with the WAIT option from within a VM Batch Facility job has resulted in a non zero return code *rcode* from the target VM Batch Facility monitor. The message that the monitor issued directly to the job owner describing the cause of the non zero return code may be identified by *mcode*.

VM Batch Facility Action: The job has been processed according to the notified message code.

User Response: This message is issued to assist with any problems that may arise with a synchronous submission from a VM Batch Facility task machine in case the original message that it identifies is not able to reach the job owner. Any remedial action is indicated by the the original message.

592E **Return code** *rc* **from** *service* **at offset** *offset* **into** *csect*

Explanation: This message documents the receipt of an unexpected return code from the named CP or CMS system service.

VM Batch Facility Action: Resulting action may be notified in a subsequent message or deduced from the return code of the VM Batch Facility function.

User Response: Correct the condition causing the failure of the system service.

593I **Job** *userid nodeid jobid* **is ready**

Explanation: This message is issued in response to a successful START command, when the command specified a job but no task machine.

VM Batch Facility Action: The job will run as soon as a task machine becomes available.

User Response: None.

594I

Explanation: This message is issued to the VM Batch Facility console to record the receipt of a BATCH START command.

VM Batch Facility Action: None.

User Response: None.

595E **Chjob** *userid nodeid jobid* **rejected; job does not recur**

Explanation: This message is issued in response to a BATCH CHJOB command, when the command specified a repeat pattern of HOLD or NOHOLD (indicating a request to suspend or resume recurrence), and the job does not recur.

VM Batch Facility Action: The command is ignored.

User Response: If you wish to hold a non recurring job then issue the BATCH CHJOB command specifying DATE HOLD.

596E   **Job/Chjob** *userid nodeid jobid* **rejected; Storage overrides for class** *classid*
           are not permitted

Explanation: The job or command has been rejected by the VM Batch Facility because an
override of the default virtual storage allocation for the job's class was specified, and the
class definition does not permit an override.

VM Batch Facility Action: The job or command is rejected.

User Response: If you are happy with the virtual storage available to the job class then do
not override it, or submit the job under a different class with adequate storage bounds.

597E   **Chjob** *userid nodeid jobid* **rejected; job cannot be made to recur**

Explanation: This message is issued in response to a BATCH CHJOB command that
would result in a non-recurring job being changed into a recurring job, and the VM Batch
Facility cannot do this. This will happen if the CHJOB command explicitly specified the
REPEAT option, or because it specified a chain condition where the chained-to job is
recurring.

VM Batch Facility Action: The command is rejected.

User Response: If you wish the job to become recurring you will need to cancel it and then
re-submit it as a recurring job.

598E   **Installation exits have not been (re)loaded**

Explanation: This message is issued when a problem occured while loading one of the
installation exits. The problem that occurred will be indicated by a preceeding DGRxxx592E
message.

VM Batch Facility Action: If the problem occurred during VM Batch Facility monitor
initialization the startup is terminated with return code 12.

If the message is issued in response to a VM Batch Facility RELOAD command the monitor
continues operating with the previous versions of all installation exits.

User Response: Correct the condition indicated by the preceeding DGRxxx592E message
and retry the command or startup.

599I   **Job** *userid nodeid jobid* **recurs implicitly**

Explanation: This message is issued in response to a BATCH SUBMIT command that
results in the job being marked as implicitly recurring because it has been chained to job that
is recurring, either explicitly or implicitly.

VM Batch Facility Action: The job is accepted.

User Response: If you do not wish the job to recur you may issuue a BATCH CHJOB
command to remove the job's chain condition, or to chain it to a job that does not recur.

600E   **You may not direct VM Batch commands to your own virtual machine**

Explanation: This message is issued in response to a VM Batch Facility command directed
at the virtual machine from which it was issued; or in response to a BATCH ADDRESS
command directing following VM Batch commands to the virtual machine at which it was
issued.

VM Batch Facility Action: The command is rejected.

User Response: Direct your VM Batch commands to a bona fide VM Batch Facility
monitor.

601I   **Job** *userid nodeid jobid* **will not recur; it is chained to a job that no longer recurs**

Explanation: This message identifies a job that has changed from an implicitly recurring job to one that does not recur, because the job that it is chained to is no longer a recurring job. Either the job at the head of the chain has been changed to not recur, or the chain has been altered so that the job now at its head is not a recurring job.

VM Batch Facility Action: The job will be dispatched according to the normal Facility rules, but it will be discarded when it completes.

User Response: If you do not want the job to run, issue the BATCH CANCEL command.

Bibliography

This bibliography lists the publications that provide information about your VM/ESA system. The VM/ESA library includes VM/ESA base publications, publications for additional facilities included with VM/ESA, and publications for VM/ESA optional features.

VM/ESA publications may be available as Adobe Portable Document Format (PDF) files, IBM BookManager® files, or printed books. For abstracts of VM/ESA publications and other library-related information, including current editions and available publication formats, see *VM/ESA: General Information*.

VM/ESA Base Publications

Evaluation

VM/ESA: Licensed Program Specifications, GC24-5744

VM/ESA: General Information, GC24-5745

Installation and Service

VM/ESA: Installation Guide, GC24-5836

VM/ESA: Service Guide, GC24-5838

VM/ESA: VMSES/E Introduction and Reference, GC24-5837

Planning and Administration

VM/ESA: Planning and Administration, SC24-5750

VM/ESA: CMS File Pool Planning, Administration, and Operation, SC24-5751

VM/ESA: Conversion Guide and Notebook, GC24-5839

VM/ESA: REXX/EXEC Migration Tool for VM/ESA, GC24-5752

VM/ESA: Running Guest Operating Systems, SC24-5755

VM/ESA: Connectivity Planning, Administration, and Operation, SC24-5756

VM/ESA: Group Control System, SC24-5757

VM/ESA: Performance, SC24-5782

Customization

IBM VM/ESA: CP Exit Customization, SC24-5672

Operation

VM/ESA: System Operation, SC24-5758

VM/ESA: Virtual Machine Operation, SC24-5759

Application Programming

VM/ESA: CP Programming Services, SC24-5760

VM/ESA: CMS Application Development Guide, SC24-5761

VM/ESA: CMS Application Development Reference, SC24-5762

VM/ESA: CMS Application Development Guide for Assembler, SC24-5763

VM/ESA: CMS Application Development Reference for Assembler, SC24-5764

VM/ESA: CMS Application Multitasking, SC24-5766

VM/ESA: REXX/VM Primer, SC24-5598

VM/ESA: REXX/VM User's Guide, SC24-5465

VM/ESA: REXX/VM Reference, SC24-5770

IBM VM/ESA: Distributed Graphical User Interface Toolkit, SC24-5724

IBM VM/ESA: Reusable Server Kernel Programmer's Guide and Reference, SC24-5852

VM/ESA: Enterprise Systems Architecture/Extended Configuration Principles of Operation, SC24-5594

VM/ESA: Programmer's Guide to the Server-Requester Programming Interface for VM, SC24-5455

VM/ESA: CPI Communications User's Guide, SC24-5595

Common Programming Interface Communications Reference, SC26-4399

Common Programming Interface Resource Recovery Reference, SC31-6821

External Security Interface (RACROUTE) Macro Reference for MVS and VM, GC28-1366

End Use

VM/ESA: CP Command and Utility Reference, SC24-5773

VM/ESA: CMS Primer, SC24-5458

VM/ESA: CMS User's Guide, SC24-5775

VM/ESA: CMS Command Reference, SC24-5776

IBM VM/ESA: Graphical User Interface Facility, SC24-5789

VM/ESA: CMS Pipelines User's Guide, SC24-5777

VM/ESA: CMS Pipelines Reference, SC24-5778

CMS/TSO Pipelines: Author's Edition, SL26-0018

VM/ESA: XEDIT User's Guide, SC24-5779

VM/ESA: XEDIT Command and Macro Reference, SC24-5780

VM/ESA: Quick Reference, SX24-5290

Diagnosis

VM/ESA: System Messages and Codes, GC24-5841

VM/ESA: Dump Viewing Facility, GC24-5853

VM/ESA: Diagnosis Guide, GC24-5854

VM/ESA: CP Diagnosis Reference, SC24-5855

VM/ESA: CP Diagnosis Reference Summary, SX24-5292

VM/ESA: CMS Diagnosis Reference, SC24-5857

Note: CP and CMS control block information is not provided in book form. This information is available on the IBM VM operating system home page (http://www.ibm.com/s390/vm).

Publications for Additional Facilities

OpenEdition® for VM/ESA

IBM OpenEdition for VM/ESA: POSIX Conformance Document, GC24-5842

IBM OpenEdition for VM/ESA: User's Guide, SC24-5727

IBM OpenEdition for VM/ESA: Command Reference, SC24-5728

IBM OpenEdition for VM/ESA: Advanced Application Programming Tools, SC24-5729

IBM OpenEdition for VM/ESA: Callable Services Reference, SC24-5726

IBM OpenEdition for VM/ESA: Sockets Reference, SC24-5741

IBM C for VM/ESA: Library Reference, SC23-3908

Debug Tool User's Guide and Reference, SC09-2137

DFSMS/VM®

VM/ESA: DFSMS/VM Function Level 221 Planning Guide, GC35-0121

VM/ESA: DFSMS/VM Function Level 221 Installation and Customization, SC26-4704

VM/ESA: DFSMS/VM Function Level 221 Storage Administration Guide and Reference, SH35-0111

VM/ESA: DFSMS/VM Function Level 221 Removable Media Services User's Guide and Reference, SC35-0141

VM/ESA: DFSMS/VM Function Level 221 Messages and Codes, SC26-4707

VM/ESA: DFSMS/VM Function Level 221 Diagnosis Guide, LY27-9589

S/390® Open Systems Adapter Support Facility for VM/ESA

Planning for the System/390 Open Systems Adapter Feature, GC23-3870

IBM VM/ESA: Open Systems Adapter Support Facility User's Guide, SC28-1992

Language Environment®

Language Environment for OS/390 & VM: Concepts Guide, GC28-1945

Language Environment for OS/390 & VM: Migration Guide, SC28-1944

Language Environment for OS/390 & VM: Programming Guide, SC28-1939

Language Environment for OS/390 & VM: Programming Reference, SC28-1940

Language Environment for OS/390 & VM: Writing Interlanguage Communication Applications, SC28-1943

Language Environment for OS/390 & VM: Debugging Guide and Run-Time Messages, SC28-1942

Publications for Optional Features

CMS Utilities Feature

VM/ESA: CMS Utilities Feature, SC24-5535

TCP/IP Feature for VM/ESA

VM/ESA: TCP/IP Function Level 320 Planning and Customization, SC24-5847

VM/ESA: TCP/IP Function Level 320 User's Guide,
SC24-5848

*VM/ESA: TCP/IP Function Level 320 Programmer's
Reference*, SC24-5849

*VM/ESA: TCP/IP Function Level 320 Messages and
Codes*, GC24-5850

VM/ESA: TCP/IP Function Level 320 Diagnosis Guide,
GC24-5851

OpenEdition Distributed Computing Environment Feature for VM/ESA

*IBM OpenEdition DCE for VM/ESA: Introducing the
OpenEdition Distributed Computing Environment*,
SC24-5735

IBM OpenEdition DCE for VM/ESA: Planning,
SC24-5737

*IBM OpenEdition DCE for VM/ESA: Configuring and
Getting Started*, SC24-5734

*IBM OpenEdition DCE for VM/ESA: Administration
Guide*, SC24-5730

*IBM OpenEdition DCE for VM/ESA: Administration
Reference*, SC24-5731

*IBM OpenEdition DCE for VM/ESA: Application
Development Guide*, SC24-5732

*IBM OpenEdition DCE for VM/ESA: Application
Development Reference*, SC24-5733

IBM OpenEdition DCE for VM/ESA: User's Guide,
SC24-5738

*IBM OpenEdition DCE for VM/ESA: Messages and
Codes*, SC24-5736

LAN File Services/ESA

Discovering LAN File Services/ESA, GK2T-5762

Introducing LAN File Services/ESA, GH24-5259

*LAN File Services/ESA: Licensed Program
Specifications*, GH24-5260

LAN File Services/ESA: VM Guide and Reference,
SH24-5264

LAN Resource Extension and Services/VM

*LAN Resource Extension and Services/VM: Licensed
Program Specifications*, GC24-5617

*LAN Resource Extension and Services/VM: General
Information*, GC24-5618

*LAN Resource Extension and Services/VM: Guide and
Reference*, SC24-5622

CD-ROM

The following CD-ROM contains PDF versions of many
VM/ESA publications and publications for some related
IBM licensed programs. It also contains all the IBM
libraries that are available in IBM BookManager format
for current VM system products and current IBM
licensed programs that run on VM/ESA.

IBM Online Library Omnibus Edition: VM Collection,
SK2T-2067

Note: Only unlicensed publications are included.

Index

Readers' Comments

VM Batch Facility
Installation, Customization, and Administration
Version 2 Release 2

Publication No. SC24-5573-02

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

> **Note**
>
> Report system problems to your IBM representative or the IBM branch office serving you.
> U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at **1-800-879-2755**.

In addition to using this postage-paid form, you may send your comments by:

FAX	1-607-752-2327		Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL		IBMLink	GDLVME(PUBRCF)

Would you like a reply? ___**YES** ___ **NO**   If yes, please tell us the type of response you prefer.

___ Electronic address: _____

___ FAX number: _____

___ Mail: (Please fill in your name and address below.)

_____   _____
Name   Address

_____   _____
Company or Organization
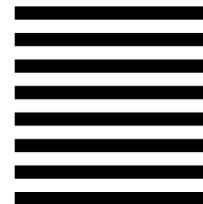
_____   _____
Phone No.

Readers' Comments
SC24-5573-02

IBM ®

Please do not staple

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60
International Business Machines Corporation
Information Development
1701 North Street
ENDICOTT NY 13760-5553

Please do not staple

SC24-5573-02

IBM®

IBM   VM Batch Facility   Install, Customize, & Administer   *Version 2 Release 2*