

# CA VM:Schedule™

## Operator Guide Version 2.0, First Edition



9/25/2013

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## CA Technologies Product References

This document references the following CA Technologies products:

- *CA Mainframe VM Product Manager*
- *CA VM:Director*
- *CA VM:Schedule User Guide* - presents concepts and procedures for tasks that end users typically perform. This guide also includes end-user commands and the format for those commands.
- *CA VM:Schedule Operator Guide* - presents concepts and procedures relevant to the day-to-day operation of CA VM:Schedule. The book also includes operator commands and the formats for those commands.
- *CA VM:Schedule Message Reference Guide* - lists all messages that CA VM:Schedule produces, and provides the following information: the possible cause of the situation that generated the message, and any actions to take in response to the situation. The book also has a cross-reference that lists the first 60 characters of each CA VM:Schedule message in alphabetical order.
- *CA VM:Schedule Administration Guide* - provides the following information:
  - Explanations of *CA VM:Schedule* and its administration
  - Instructions for customizing *CA VM:Schedule* to fit site-specific requirements
  - Descriptions of any special administrative operations that must be performed
- *CA VM:Schedule Installation Guide*. - gives information about installing and maintaining *CA VM:Schedule*.

Other CA product guides that are referenced in this book and to which you can refer include:

- *CA VM:Schedule Installation Guide*
- *CA Mainframe VM Product Manager Reference Guide*
- *CA Mainframe VM Product Manager Interface Guide*
- *CA Mainframe VM Product Manager Generalized Report Writer Reference Guide*
- *CA VM:Archiver System Administration Guide*
- *CA VM:Batch Administration Guide*

# Contact CA Technologies

## Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

## Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to [techpubs@ca.com](mailto:techpubs@ca.com).

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

# Documentation Changes

## CA VM:Schedule 2.0, First Edition, 9/25/2013

The following documentation updates have been made since the last release of this documentation:

- Global change—All references to the user ID for system administrator and maintenance functions changed from VMRMAINT to VMANAGER.
- Global change—Names and branding of related products were updated.
- [Format Conventions for Code Syntax](#) (see page 11)—Changed to new common standards.
- [QLOCK Command](#) (see page 48)—New command that displays all locks being held.
- [QPCB Command](#) (see page 49)—New command that lists all active processes and the commands that they are executing.
- [QREPLY Command](#) (see page 50)—New command that lists all processes that are awaiting a reply.
- [REPLY Command](#) (see page 51)—Command changed to reflect transfer of query functionality to new QREPLY command.
- [DISPLAY Command](#) (see page 42)—SYSNAME option added to command.



# Contents

---

<b>Chapter 1: About This Book</b>	<b>11</b>
Format Conventions for Code Syntax.....	11
<b>Chapter 2: About CA VM:Schedule</b>	<b>17</b>
Authorizations.....	18
Executing CA VM:Schedule Operator Commands.....	18
Help with CA VM:Schedule Commands .....	19
Help With Messages.....	20
<b>Chapter 3: Controlling CA VM:Schedule Operation</b>	<b>21</b>
Listing Request Information (DISPLAY).....	22
Full-Screen Example .....	23
Line-Mode Examples .....	23
Listing Run Times (WHEN).....	24
Full-Screen Example .....	24
Line-Mode Examples .....	25
Starting and Stopping Request Processing (INITIATE).....	26
Full-Screen Example .....	26
Line-Mode Examples .....	27
Changing the User ID On Which a Request Runs (TRANSFER) .....	28
Clearing Invalid Password Counts (RESET) .....	28
Full-Screen Example .....	28
Line-Mode Examples .....	29
Managing Request Classes (CLASS) .....	30
Full-Screen Example .....	30
Line-Mode Examples .....	31
Issuing CMS Subset Commands on CA VM:Schedule (CMS) .....	31
Responding to CA VM:Schedule System Messages (REPLY).....	32
Line-Mode Examples .....	32
Shutting Down CA VM:Schedule (END).....	32
Line-Mode Examples .....	33
Suspending and Resuming CA VM:Schedule (SUSPEND and RESUME).....	33
Line-Mode Examples .....	33

---

## Chapter 4: Command Reference

35

ABEND Command.....	37
CLASS Command .....	38
Definitions .....	38
Examples .....	39
CMS Command.....	41
Definitions .....	41
Example.....	41
DISPLAY Command.....	42
Definitions .....	42
Examples .....	43
END Command .....	44
INITIATE Command .....	45
Definitions .....	45
Examples .....	45
OPERATOR Command .....	47
QLOCK Command.....	48
Description .....	48
QPCB Command .....	49
Description .....	49
QREPLY Command.....	50
Example.....	50
REPLY Command .....	51
Definitions .....	51
Examples .....	51
RESET Command .....	52
Definitions .....	52
Description .....	52
Examples .....	52
RESUME Command .....	53
SUSPEND Command.....	54
TRANSFER Command .....	55
Definitions .....	55
Restrictions .....	55
Examples .....	55
WHEN Command.....	56
Definitions .....	56
Output Destinations.....	58
Controlling the Number of Runs Displayed.....	58
Listing Many Requests .....	58
Examples .....	58





# Chapter 1: About This Book

---

The *CA VM:Schedule Operator Guide* contains task-oriented information to teach new operators how to control *CA VM:Schedule* operation. This guide also contains a command reference. The command reference allows experienced operators to look up the information they need and get on with their work.

This section contains the following topics:

[Format Conventions for Code Syntax](#) (see page 11)

## Format Conventions for Code Syntax

This section describes the format and conventions used to document commands, utilities, and user exits. Each convention provides examples, describing how to use commands, how to use options, or how the system responds to user entries.

**Note:** The examples and instructions throughout this document use VMANAGER as the user ID for system administrator and maintenance functions. Also, this document uses VMSCHED as the user ID for the service virtual machine (svm). These user IDs are the default values. If you use non-default user IDs for the system administrator or the svm, replace the default values throughout this document with the values that you use.

### Command Abbreviations

When a command contains uppercase and lowercase letters, then the uppercase letters denote the shortest acceptable abbreviation that you can use to type the command. However, when a code item appears entirely in uppercase letters, you cannot abbreviate the item.

You can type the code item in uppercase letters, lowercase letters, or any combination.

#### Example:

`CMDName`

In this example, you can enter `CMDNA`, `CMDNAM`, or `CMDNAME` in any combination of uppercase and lowercase letters.

### Continuation

The code syntax or code fragment definitions can continue from one line to the next line. The following examples describe code continuation:

**Example 1:**

```
A | B C | D
```

This code is equivalent to the following code:

```
A  
| B C  
| D
```

**Example 2:**

```
{choice1 | choice2 | choice99}
```

This code is equivalent to the following code:

```
{ choice1  
| choice2  
| choice99 }
```

**Default Values**

An underlined code item denotes the default value. The system uses the default value unless you override it. You can override the default value by coding an option from the available list.

**Example:**

```
[parm1 | parm2 | parm3]
```

In this example, the code item *parm1* is the default value, and this is used by the system when you do not specify any of the options. However, you can code *parm1*, *parm2*, or *parm3*.

**Keywords and Constants**

A keyword name or constant always appears in uppercase letters. Code the keyword name or constant exactly as shown in the following example:

**Example:**

```
STOP {tracenum | * | [USER] userid}
```

This example displays the USER keyword.

**Optional Choices**

Defines optional code items—denoted by square brackets around a code item.

**Example:**

```
CMDName [parm1]
```

In this example, you can choose *parm1* or no parameter at all. However, when two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

**Multiple Optional Choices**

When two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

**Example:**

In this example, you can choose *parm1*, *parm2*, *parm3*, or nothing at all.

```
[parm1 | parm2 | parm3]
```

**Positional Parameters**

Commands with positional parameters are identified by nested square brackets. Each positional parameter requires the specification of all previous positional parameters. The following example describes the positional parameter:

**Example:**

```
CMDName [posparm1 [posparm2 [posparm3]]]
```

In this example, *posparm3* can be specified only when *posparm1* and *posparm2* are also specified.

**Repeatable Choices**

A list of code items enclosed in square brackets and followed by an ellipsis means that you can select more than one item or, in some cases, repeat a single item.

**Example:**

```
[value1 | value2 | value3] ...
```

In this example, you can choose a single value, more than one value, or none of the values.

**Repetition**

An ellipsis following a code item means that the code item can be repeated.

**Example:**

```
Repeat...
```

### Required Choices

You must select one item from a list of items when they are enclosed in curly braces. The items are separated by a vertical bar character.

#### Example:

```
CMDName {A | B | C}
```

In this example, your choice results in CMDNAME A, CMDNAME B, or CMDNAME C.

### Special Symbols

The following list describes the meaning of the special symbols used in codes:

- {} (encloses a list of operands, one of which is required).
- [] (enclose an optional operand or operands).
- " " (enclose the name of a syntax fragment)
- (underline (identifies a default value)
- | (separates alternative operands)
- . . . (Indicates that the preceding item or group can be repeated).

### Symbols

The following list displays symbols. These symbols should be coded exactly as they appear in the code syntax.

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

**Syntax Fragments**

Some codes use fragments, when the code syntax is too lengthy. The fragment name appears between double quotes in the code syntax.

The expanded fragment appears in the syntax after all other parameters or at the bottom of the code syntax. A heading with the fragment name identifies the expanded fragment.

**Example:**

```
CMDName "Parms"  
Parms :  
[A_ | B | C]
```

In this example, the fragment is named "Parms", and the expanded fragment appears at the bottom of the code syntax.

**System Response**

Uppercase characters represent system responses or prompts.

**Example:**

```
ENTER YOUR LOGON PASSWORD:
```

This example displays a system response.

### User-Entered Commands or Records

User-entered commands are shown in lowercase letters even though you can enter commands in either upper or lower case.

#### Example:

```
vmsecure addentry writers tcom (noformat nowait
```

This example shows what a user-entered command looks like.

In this example "writers tcom" is the file name and file type of the directory entry you are adding.

However, if the entry is a record, it appears in uppercase letters.

#### Example:

```
ACCESS DRCT 1B0 U
```

This example shows a configuration record that is entered by a user.

### Variables

Lowercase items in italics denote variables.

#### Example:

```
CMDName varname
```

In this example, *varname* represents a variable that you must specify when you code the command CMDNAME.

# Chapter 2: About CA VM:Schedule

---

The *CA VM:Schedule* product, referred to as *CA VM:Schedule* in this guide, is a component of the *CA VM:Manager* suite of products. This component allows users to schedule requests to process in the future.

A scheduled request consists of an EXEC file or command and the scheduling instructions. *CA VM:Schedule* requests can print SCRIPT files, compile COBOL programs, issue CP and CMS commands, and run other types of CMS programs. Users can schedule requests to run just once or to repeat at regular intervals. For example, a request can be scheduled to run at the following times:

- 1:00 p.m. this coming Friday.
- 6:00 a.m. every business day.
- On the last day of every month.

Users can schedule requests to run on their own user IDs or the user IDs of other users. Alternatively, they can schedule EXECs to run on the *CA VM:Schedule* service virtual machine.

Most scheduled requests run on the requesting user's user ID. When it is time for the request to run, *CA VM:Schedule* checks to see if the user is logged on. If so, the product reminds the user to log off. Once the user ID is logged off, *CA VM:Schedule* autologs it—that is, logs it on in disconnected mode. The request then runs as if the user entered the command directly from the terminal.

*CA VM:Schedule* monitors and records what happens during request processing. It sends a notification file to the user's reader in the following circumstances:

- When the request runs for the last time.
- Any time an error occurs within the request.

Your site can use *CA VM:Schedule* to set up frequently repeated work, such as monthly accounting reports, in advance. Users can schedule time-consuming tasks to run when they are not in the office so that the requests do not interrupt other work. Users can automatically print log every Friday, run DB2 reports each morning before they get in, and remind colleagues about meetings. Programmers can compile and test programs overnight. Data center staff can automate system backups, performance reporting, and system control utilities.

This section contains the following topics:

[Authorizations](#) (see page 18)

[Executing CA VM:Schedule Operator Commands](#) (see page 18)

[Help with CA VM:Schedule Commands](#) (see page 19)

[Help With Messages](#) (see page 20)

## Authorizations

You must be authorized as an operator in the *CA VM:Schedule* system to perform the tasks and use the commands that this book describes. Contact your *CA VM:Schedule* system administrator if you experience difficulties.

## Executing CA VM:Schedule Operator Commands

To execute operator commands, access *CA VM:Schedule* with authorization as an OPERATOR. You can access the product in full-screen menus or in line mode. The screens are easier to use, especially if you are unfamiliar with *CA VM:Schedule* or you do not schedule requests often. The line-mode commands let you use the product within EXECs or from programs.

To open the full-screen menu, enter the following command from CMS:

```
vmsched oper
```

**Note:** VMSCHED is the default name of the *CA VM:Schedule* service virtual machine. If the *CA VM:Schedule* service virtual machine at your site has a different name, use that name instead.

**Note:** When you first access *CA VM:Schedule*, a copyright screen can appear. To clear the screen, press Enter or wait 15 seconds.

After the copyright screen clears, the Operator Main Menu displays.

```

-----
                                Operator Main Menu                                VM:Schedule
-----
To choose one of the following, type its number and press ENTER.

    1 DISPLAY  Display pending initiations.
    2 WHEN     Display future initiation(s) for a request.
    3 INITIATE Control initiation of scheduled requests.
    4 TRANSFER Transfer request ownership to another userid.
    5 RESET    Reset a password count to zero.
    6 CLASS    Control request classes.
    7 CMS      Invoke a CP or CMS command on the VM:Schedule machine.
    8 REPLY    Respond to a VM:Schedule message.
    9 END      Terminate VM:Schedule system operation.
   10 SUSPEND Suspend VM:Schedule operation temporarily.
   11 RESUME   Resume a suspended VM:Schedule machine.
   12 ABEND   Force VM:Schedule to end abnormally.

                                Copyright (c) 2013 CA. All rights reserved.
-----
PF: 1 Help    2 ...    3 End    4 Return  5 ...    6 ...
PF: 7 ...    8 ...    9 ...   10 Print  11 ...   12 ...

==>
-----

```

*CA VM:Schedule* displays the main menu for your user level. Type the number of the task you want to perform and press Enter. If you prefer, you can enter the command name or its abbreviation instead on its menu's or screen's command line. *CA VM:Schedule* displays the fill-in-the-blank screen for that task. Where appropriate, default values for fields are shown. When you complete the task, press PF2 to submit your instructions to *CA VM:Schedule*. If you forget to fill in a required field, *CA VM:Schedule* prompts you to enter the information.

If you want further explanation for any item on a screen, press Tab to put your cursor on the item and press PF1. *CA VM:Schedule* displays an explanation.

To leave *CA VM:Schedule* full-screen menu processing, press PF3 from the command screens. You return to the main menu. Press PF3 again until you return to CMS.

For more information about full-screen menus, see the *User Guide*.

To execute a command in line mode, enter the name of the service virtual machine (default is **VMSCHED**) followed by the command name and any operands. When a line mode command completes, a return code is presented to the issuing user.

For example, to transfer the ACCOUNTS request from user ALICE to user LOUISE through a line-mode command, enter from CMS:

```
vmsched transfer accounts alice louise
```

Any error messages are edited according to your virtual machine's current EMSG setting.

The complete formats for operator line-mode commands are listed in [Command Reference](#) (see page 35).

## Help with CA VM:Schedule Commands

You can review explanations of *CA VM:Schedule* commands at your terminal. To get help for commands, from CMS enter:

```
help vmsched menu
```

A help menu displays. Move the cursor to the command you need help with and press Enter.

## Help With Messages

*CA VM:Schedule* provides files for the CMS HELP facility. These files contain information about *CA VM:Schedule* commands and system messages. The information in these files matches the information in the documentation. The HELP files include complete descriptions of messages that are listed but not explained here.

The *CA VM:Schedule Messages Reference Guide* contains the following information:

- All messages that *CA VM:Schedule* can generate
- The possible causes of each message
- How to respond to that message

For more information about a system message you receive, refer to the *CA VM:Schedule Messages Reference Guide* or use the CMS HELP facility for that message.

To use the CMS HELP facility for a *CA VM:Schedule* message, type **help** followed by a space. Then type a string that is composed of the following parts:

- The three-character product identifier (**vmd**)
- The message number
- The severity code

Do not type the three-letter code that is displayed after the product identifier. This code is the routine or module that issued the message.

### Example: Getting help on the message VMDCOM1091E

To get help on the message VMDCOM1091E, enter the following command from CMS:

```
help vmd1091e
```

When you have finished reading the message help, press PF3 to exit the CMS HELP facility.

For more information about using the CMS HELP facility, refer to the CMS User's Guide for your system.

# Chapter 3: Controlling CA VM:Schedule Operation

---

This chapter describes common tasks used in the day-to-day operation of CA VM:Schedule.

This section contains the following topics:

[Listing Request Information \(DISPLAY\)](#) (see page 22)

[Listing Run Times \(WHEN\)](#) (see page 24)

[Starting and Stopping Request Processing \(INITIATE\)](#) (see page 26)

[Changing the User ID On Which a Request Runs \(TRANSFER\)](#) (see page 28)

[Clearing Invalid Password Counts \(RESET\)](#) (see page 28)

[Managing Request Classes \(CLASS\)](#) (see page 30)

[Issuing CMS Subset Commands on CA VM:Schedule \(CMS\)](#) (see page 31)

[Responding to CA VM:Schedule System Messages \(REPLY\)](#) (see page 32)

[Shutting Down CA VM:Schedule \(END\)](#) (see page 32)

[Suspending and Resuming CA VM:Schedule \(SUSPEND and RESUME\)](#) (see page 33)

## Listing Request Information (DISPLAY)

You can use the DISPLAY command to list information about requests scheduled to run in the future. The DISPLAY command shows only the next initiation of a request. (Use the WHEN command to display more than one run time for a request. See the [Listing Run Times](#) (see page 24) section.)

The list that the DISPLAY command produces includes the following requests:

- Run time
- Request name
- Resource limits
- User ID
- Command for requests that are submitted through *CA VM:Schedule*.

These requests are listed in the order that *CA VM:Schedule* received them. You can limit the list by the following criteria:

- Requests with similar names
- Requests that run on a specific user ID
- Requests that run in a particular time range.

DISPLAY does not cover the following requests:

- Requests scheduled to be released
- Requests in classes
- Requests that have already reached their scheduled time and are trying to autolog a user ID

## Full-Screen Example

To use the full-screen DISPLAY command, select DISPLAY from the Operator Main Menu and press ENTER. The Display Pending Initiations screen appears.

To list the next initiation of all requests belonging to user ID LOUISE whose next run is scheduled between April 1, 20YY and June 1, 20YY, fill out the screen as shown in the following illustration.

```

                                Display Pending Initiations                                VM:Schedule
-----
To display the initiations pending for one or more userids, type the
information below or simply press ENTER for default display.

    Userid: LOUISE__   Request Name: *______

                                OPTIONS

    Display initiations FROM (mm/dd/yy): 04/01/YY
    Display initiations TO (mm/dd/yy):   06/01/YY

-----
PF: 1 Help      2 ...      3 End      4 Return  5 ...      6 ...
PF: 7 ...      8 ...      9 ...     10 Print 11 ...     12 Submit
==>
-----

```

## Line-Mode Examples

- To list the next initiation of all requests belonging to userid LOUISE whose next run is scheduled to run between April 1 and June 1, enter the following command:
 

```
vmsched display louise * (from 04/01/YY until 06/01/YY)
```
- To list the next initiation of all requests named REPORT scheduled to run today, enter the following command:
 

```
vmsched display * report
```
- To list the next initiation of all requests whose names begin with REP and whose next initiation falls between 06/24/YY and 06/26/YY, enter the following command:
 

```
vmsched display * rep* (from 06/24/YY until 06/26/YY)
```
- To list the next initiation of all requests belonging to all userids that are scheduled for today, enter the following command:
 

```
vmsched display * *
```

## Listing Run Times (WHEN)

You can use the WHEN command to find out when a scheduled request runs. You can check on a request's time of next run, next several runs, or all runs within a given date range. You can specify up to 9,999 runs.

If you do not specify the number of requests to check, *CA VM:Schedule* lists up to 999 runs for the specified date range specified. If you do not specify the range of dates either, *CA VM:Schedule* lists the next 999 runs starting today.

The WHEN feature performs the following functions:

- Display pending requests for the remainder of the day
- Display requests for multiple user IDs
- List requests scheduled to run this weekend when the system is down for maintenance

### Full-Screen Example

To use the full-screen WHEN command, select WHEN from the Operator Main Menu and press ENTER. The Display Future Initiation(s) of a Request screen appears, as shown in the following figure.

To list all the requests for user ID DIANE scheduled to run during the weekend of June 6 through June 8, fill out the screen as follows:

```

                                Display Future Initiation(s) of a Request          VM:Schedule
-----
To display information about future initiations of a request, type the
information below and press ENTER.

Request Name:  *______  For Userid: DIANE  (Password:      )
From (mm/dd/yy): 06/06/YY  Until (mm/dd/yy): 06/08/YY
Display next: 999  initiations  Sorted by date (Y/N)? N

Specify an option below or simply press ENTER for a full-screen display.

Output to printer (Y/N)?  N
Output to disk (filename)? _____

-----
PF: 1  Help    2  ...    3  End    4  Return  5  ...    6  ...
PF: 7  ...    8  ...    9  ...   10  Print  11  ...   12  Submit
====>
-----

```

When you press ENTER, the screen displays a list of the requests to display during the specified dates, like the following figure.

Display Future Initiation(s) of a Request							VM:Schedule
Request: TEST		User: DIANE	From: 06/06/YY	Until: 06/08/YY	Next: 999		
Number	Request	Userid	Day	Date	Time	Status	
1	TEST	DIANE	FRI	06/06/YY	08:29:32		
2	TEST	DIANE	MON	06/07/YY	08:29:32		
3	TEST	DIANE	TUE	06/08/YY	08:29:32		
-----							
PF: 1	Help	2 ...	3 End	4 Return	5 ...	6 ...	
PF: 7	Backward	8 Forward	9 ...	10 Print	11 ...	12 ...	
=>							
-----							

## Line-Mode Examples

- To list the requests for user DIANE that are scheduled to run over a weekend (June 6 through June 8) when the system is scheduled to be down for maintenance, enter the following command:

```
vmsched when * (user diane from 06/06/YY until 06/08/YY sort
```

- To list the next ten runs of the REPORT request that is scheduled to run today, enter:

```
vmsched when report (next 10
```

To avoid tying up your terminal, schedule the WHEN command to run when you do not need your user ID for interactive work.

- To display all requests that are scheduled to run on November 26 and 27, place the following command in an EXEC and schedule the EXEC to run later:

```
vmsched when * (from 11/26/YY until 11/27/YY user * print sort
```

## Starting and Stopping Request Processing (INITIATE)

To pass the following instructions to *CA VM:Schedule*, use the INITIATE command:

- Start processing all requests.
- Process only the requests that process on *CA VM:Schedule*.
- Stop processing all requests or all except those requests that run on *CA VM:Schedule*.

If the system goes down and it is brought back up later, INITIATE commands all remain in effect. Each time one of these commands is issued, a file named VMSCHED INITIATE, which contains the command, is created on the *CA VM:Schedule* 1B0 disk. If the 1B0 disk is not available, the file is written to the service virtual machine's A-disk. If *CA VM:Schedule* is shut down and later reinitialized, it checks for this file. If the file does not exist, all requests start normally. If the file does exist, requests start as specified in this command.

Use this feature to stop *CA VM:Schedule* operation temporarily during periods of peak resource consumption without bringing the system down.

### Full-Screen Example

To use the full-screen INITIATE command, select INITIATE from the Operator Main Menu and press ENTER. The Control Initiations of Scheduled Requests screen appears.

The following illustration shows how to fill out the screen to stop all request processing.

```

Control Initiation of Scheduled Requests          VM:Schedule
-----
To set initiations on or off, type the information below and press ENTER.
Initiations that are set off are rescheduled whenever they are set on.

Set all initiations on (Y/N):                  -
Set only VM:Schedule initiations on (Y/N):    -
Set all initiations off (Y/N):                 Y
Set all initiations off except VM:Schedule (Y/N): _

-----
PF: 1 Help    2 ...    3 End    4 Return  5 ...    6 ...
PF: 7 ...    8 ...    9 ...   10 Print 11 ...   12 Submit
==>
-----

```

## Line-Mode Examples

- To stop all request processing, enter the following command:  
`vmsched initiate off all`
- To stop processing for all requests except those that run on *CA VM:Schedule*, enter the following command:  
`vmsched initiate off except vmd`
- To start all request processing, enter the following command:  
`vmsched initiate on all`
- To start processing only requests that run on *CA VM:Schedule*, enter the following command:  
`vmsched initiate on only vmd`

## Changing the User ID On Which a Request Runs (TRANSFER)

To change the user ID on which a request runs, use the TRANSFER command. You must have TRANSFER authorization to run this command. Use this feature to reassign requests belonging to an employee who left the company or changed departments. TRANSFER is available in both full-screen and line-mode.

*CA VM:Schedule* checks for duplicate request names. You cannot transfer a request to another user who has a request of the same name pending. If you attempt to transfer a request that would result in a duplicate name, *CA VM:Schedule* rejects the transfer. In this case, the product returns an error message and a CMS return code of 24.

- To transfer all of user ID HARRY's requests to user ID DALE, enter:  
`vmsched transfer * harry dale`
- To transfer the ACCOUNTS request from user ID ALICE to user ID LOUISE, enter:  
`vmsched transfer accounts alice louise`

## Clearing Invalid Password Counts (RESET)

You can use the RESET command to clear the counter of invalid passwords a user has entered on *CA VM:Schedule* commands and screens. You can clear the counter for invalid entries a user has made for another user, or for all other users on the system.

Invalid password entries that are made by and for all users are automatically cleared when *CA VM:Schedule* initializes.

RESET can be used in both full-screen and line mode. Examples of both are provided.

### Full-Screen Example

User JOHANN has entered too many invalid passwords for user ID MARCIA. Before JOHANN can use *CA VM:Schedule* again, you must clear his invalid password count against user ID MARCIA.

To access the full-screen RESET command, select RESET from the Operator Main Menu and press ENTER. The Reset a Password Count to Zero screen appears.

Fill out the screen to clear the invalid entries user JOHANN has made trying to enter the password for user MARCIA.

```

                                Reset a Password Count to Zero                                VM:Schedule
-----
To reset a user's invalid password count to zero, type the information
below and press ENTER.

    Userid who entered
    the invalid passwords:          JOHANN__

    Userid of the virtual machine
    owning the correct password:    MARCIA__

-----
PF: 1 Help      2 ...      3 End      4 Return  5 ...      6 ...
PF: 7 ...      8 ...      9 ...     10 Print 11 ...     12 Submit
==>
-----

```

Sample Input: Reset a Password Count to Zero

## Line-Mode Examples

- To clear the invalid entries JOHANN has made trying to enter the password for MARCIA, enter this command:  
`vmsched reset johann marcia`
- To clear the invalid entries JOHANN has made trying to enter the password for any user on the system, enter this command:  
`vmsched reset johann *`

**Note:** The maximum number of attempts a user can make to schedule a request that requires a password is specified on the JOURNAL record in the VMSCHED CONFIG file.

## Managing Request Classes (CLASS)

Use the CLASS command to list status information about request classes, and to start and stop class processing. You can also start a single request in a class. Use this feature to:

- Check class use to see if you need to adjust resource and autolog limits
- Start a critical request when the system is too busy to run the whole class

### Full-Screen Example

To use the full-screen CLASS command, select CLASS from the Operator Main Menu and press ENTER. The Control Request Classes screen appears.

Fill out the screen to display requests scheduled to run in class A between June 1 and 30.

```
Control Request Classes VM:Schedule
-----
To control classes currently defined in the VM:Schedule CLASS configuration
file record, type the information below and press ENTER.

Display Class: A OPTION: Beginning: 06/01/YY
                  Ending: 06/30/YY

Start Class:  _ OPTION: Stop class automatically (Y/N)?  _
                  Change maximum autologged machines to:  __

Stop Class:  _

Drain Class  _

Run Class Job Number:  _____ From Class:  _

-----
PF: 1 Help    2 ...    3 End    4 Return  5 ...    6 ...
PF: 7 ...    8 ...    9 ...   10 Print 11 ...   12 Submit
==>
```

## Line-Mode Examples

- To list requests in all classes, enter:  

```
vmsched class display *
```
- To list requests scheduled to run in class A between July 1 and 31, enter:  

```
vmsched class display a (from 07/01/YY until 07/31/YY)
```
- To start processing a single request, the one CA VM:Schedule assigned REQID 34, in class A, enter:  

```
vmsched class run a 34
```
- To start processing all requests in class A, in the order they are queued, enter:  

```
vmsched class start a (autostop maxstart 12)
```

AUTOSTOP is optional; it automatically keeps any more requests from running in the class after the ones now in the queue have run. MAXSTART 12 temporarily changes the number of userids the class can autolog at one time to 12. When the class is stopped, the number is reset to the default given in the CLASS record in the VMSCHED CONFIG file.
- To process only class A requests that are running now and requests that are scheduled to run now but still trying to autolog their userids, enter:  

```
vmsched class drain a
```

All other requests are kept in the queue to run next time class A is started.
- To process only class A requests that are running now, enter:  

```
vmsched class stop a
```

Class A requests that are scheduled to run now but are still trying to autolog are put back in the queue to run next time class A is started.

## Issuing CMS Subset Commands on CA VM:Schedule (CMS)

You can use the CMS command to execute CMS subset commands on the *CA VM:Schedule* service virtual machine. Execute this command from the Operator Main Menu by selecting CMS. Alternatively, use the line-mode format that is shown in the following example.

To issue a CMS QUERY DISK command to *CA VM:Schedule*, enter:

```
vmsched cms query disk
```

The output from this command displays on the *CA VM:Schedule* service virtual machine, not on the user ID issuing the CMS command.

## Responding to CA VM:Schedule System Messages (REPLY)

To list the *CA VM:Schedule* messages that are waiting for replies, use the QREPLY command. To reply to one of those messages, use the REPLY command. Execute these commands from the Operator Main Menu by selecting REPLY. Alternatively, use the line-mode format that is shown in the following examples.

**Note:** QREPLY does not appear as an option on the *CA VM:Schedule* operator menu. The command is available on the REPLY screen under the following selection:

Display all messages awaiting an operator reply (Y/N):

### Line-Mode Examples

- To list all the *CA VM:Schedule* messages waiting for responses and the ID numbers for the replies, enter:

```
vmsched qreply
```

- The CLOCKCHK record in the VMSCHED CONFIG file controls the use of the *CA VM:Schedule* internal clock.

During system initialization, *CA VM:Schedule* compares the CPU clock to its internal clock. If the CPU clock is either ahead of or behind the internal *CA VM:Schedule* clock by the amount specified in the CLOCKCHK record, *CA VM:Schedule* sends system messages 0103W and 0104W to its system operator console. These messages warn you about the time difference, and allow you to double check the *CA VM:Schedule* clock. If the clock is correct and you reply OK to message 0104W, *CA VM:Schedule* executes the SKIP and CANCEL commands on requests whose WITHIN time expired while *CA VM:Schedule* was shut down.

If the CPU clock is correct, enter the following command to continue *CA VM:Schedule* initialization:

```
vmsched reply 01 ok
```

This command causes *CA VM:Schedule* to adjust its own clock file, complete initialization, and reschedule any requests appropriately, thus making up for the time the system was not operating.

If you determine that the CPU clock is incorrect, follow the procedures used at your site to set the CPU clock correctly. *CA VM:Schedule* reinitializes when you bring the system up.

## Shutting Down CA VM:Schedule (END)

To shut down *CA VM:Schedule* operation completely, use the END command. Requests that are scheduled to run or are in the process of starting do not start. You can shut down *CA VM:Schedule* while the service virtual machine is suspended.

Use this feature to:

- Prepare for upgrades
- Perform maintenance to your VM system or *CA VM:Schedule*
- Shut down and dump virtual storage

The END command is available in both full-screen and line mode.

## Line-Mode Examples

- To shut down CA VM:Schedule, enter:  

```
vmsched end
```

To restart the system, you must reinitialize CA VM:Schedule.
- To shut down CA VM:Schedule and dump virtual storage as specified in the DUMP record in the VMSCHED CONFIG file, enter:  

```
vmsched abend
```

## Suspending and Resuming CA VM:Schedule (SUSPEND and RESUME)

To stop *CA VM:Schedule* operation without shutting down the system, use the SUSPEND command. Use this feature in order to:

- Run system backups
- Run the VMDRPT utility

When you are ready to resume *CA VM:Schedule* system operation, use the RESUME command. Alternately, if you must end *CA VM:Schedule* (shut it down) while it is suspended, follow the SUSPEND command with the END command.

Both SUSPEND and RESUME are available in full-screen and line mode.

## Line-Mode Examples

- To temporarily stop all CA VM:Schedule operations so that no CA VM:Schedule commands except RESUME and END can be processed, enter:  

```
vmsched suspend
```
- To restart CA VM:Schedule operation after you previously stopped it with SUSPEND, enter:  

```
vmsched resume
```



# Chapter 4: Command Reference

---

The following table summarizes the *CA VM:Schedule* operator commands. Unless otherwise noted, you must have authorization as an OPERATOR to use these commands. For complete information about each command, see the command descriptions following the table.

Command	Function
ABEND	Shuts down <i>CA VM:Schedule</i> and dumps virtual storage
CLASS	Initiates and terminates processing of classed requests or displays information about classed requests
CMS	Executes a CMS subset or CP command on <i>CA VM:Schedule</i>
DISPLAY	Lists the next initiation of scheduled requests, but excludes class requests, the RELEASE requests, and requests already trying to start
END	Shuts down <i>CA VM:Schedule</i>
INITIATE	Controls request initiations by establishing periods of time during which no requests initiate
OPERATOR	Displays the main menu for the operator screens for <i>CA VM:Schedule</i>
QREPLY	Lists any messages that are waiting for a reply
REPLY	Communicates a response to <i>CA VM:Schedule</i>
RESET	Clears the invalid password count for a user
RESUME	Restarts <i>CA VM:Schedule</i> after the SUSPEND command has stopped it
SUSPEND	Suspends <i>CA VM:Schedule</i> activity temporarily
TRANSFER	Changes the user ID on which a request runs
WHEN	Lists the times that a request runs

This section contains the following topics:

- [ABEND Command](#) (see page 37)
- [CLASS Command](#) (see page 38)
- [CMS Command](#) (see page 41)
- [DISPLAY Command](#) (see page 42)
- [END Command](#) (see page 44)
- [INITIATE Command](#) (see page 45)
- [OPERATOR Command](#) (see page 47)
- [QLOCK Command](#) (see page 48)
- [QPCB Command](#) (see page 49)
- [QREPLY Command](#) (see page 50)
- [REPLY Command](#) (see page 51)
- [RESET Command](#) (see page 52)
- [RESUME Command](#) (see page 53)
- [SUSPEND Command](#) (see page 54)
- [TRANSFER Command](#) (see page 55)
- [WHEN Command](#) (see page 56)

## ABEND Command

Use the ABEND command to force an abnormal termination (abend code ABN001) of *CA VM:Schedule* and create a dump. Control returns to CMS with a return code of 200 following the abend.

ABEND

## CLASS Command

Use the CLASS command to initiate and terminate processing of classed requests or to display information about classed requests.

```
Class {"Parms"}
```

Parms:

```
    DISPLAY [class | *] [{"Date range options"}]  
| DRAIN class  
| RUN class requestnumber  
| START class [{"Start options"}]  
| STOP class
```

Date range options:

```
[FRom mm/dd/yy]  
[UNtil mm/dd/yy]
```

Start options:

```
[AUTOSTOP]  
[MAXSTART number]
```

## Definitions

### **DISPLAY *class***

Lists status of requests in the indicated class. The asterisk instead of a specific class name displays all defined classes. The asterisk (\*) is the default.

### **DRAIN *class***

Stops all requests not yet started or trying to start in the indicated class.

### **RUN *class requestnumber***

Starts one run of request *requestnumber* in the indicated class.

### **START *class***

Starts all requests in the specified class.

### **STOP *class***

Stops all requests not yet started in the indicated class, including requests trying to start.

**FROM *mm/dd/yy***

Defines the earliest date of the requests you want to display. This date can be in the present or future.

**UNTIL *mm/dd/yy***

Defines the latest date of the requests you want to display. This date can be in the present or future.

**AUTOSTOP**

Prevents any more requests from running in the class after the ones now in the queue have run.

**MAXSTART *number***

Defines the maximum number of requests that can be autologged at one time.

## Examples

- To list requests in all classes, enter the following command:  

```
vmsched class display *
```
- To list requests that are scheduled to run in class A between June 1 and 30, enter the following command:  

```
vmsched class display a (from 06/01/YY until 06/30/YY
```
- To start processing a single request, the one *CA VM:Schedule* assigned REQID 34, in class A, enter the following command:  

```
vmsched class run a 34
```
- To start processing all requests in class A, in the order they are queued, and to keep any more requests from running in the class after the ones now in the queue have run, and temporarily to change the number of userids the class can autolog at one time to 12, enter the following command:  

```
vmsched class start a (autostop maxstart 12
```

When the class is stopped, the number is reset to the default given in the CLASS record in the VMSCHED CONFIG file.

- To process only class A requests that are running now and requests that are scheduled to run now but still trying to autolog their userids, enter the following command:

```
vmsched class drain a
```

All other requests are kept in the queue to run next time that class A is started.

- To process only class A requests that are running now, enter the following command:

```
vmsched class stop a
```

Class A requests that are scheduled to run now but are still trying to autolog are put back in the queue to run next time that class A is started.

## CMS Command

To execute a CMS subset or CP command on the *VM Schedule* service virtual machine, use the CMS command.

Any output that this command generates appears on the Schedule service virtual machine console. The output does not appear on the console of the user ID that issues the CMS command.

```
CMS {command | CP command}
```

## Definitions

### ***command***

Defines a CMS subset command, including any necessary parameters.

### ***CP command***

Defines a CP command, including any necessary parameters.

## Example

To issue a CMS QUERY DISK command to *CA VM:Schedule*, enter the following command:

```
vmsched cms query disk
```

## DISPLAY Command

To list the next initiation of scheduled requests, use the DISPLAY command. This command excludes class requests, RELEASE requests, and requests that are already trying to start.

```
DISplay [userid | * ] [requestname | *] [{"Options"}]
```

Options:

```
[FRom mm/dd/yy]  
[UNtil mm/dd/yy]  
[SYSNAME {* | systemname}]
```

## Definitions

### ***userid***

Defines the submitting user ID of the request for which you want to display information. An asterisk instead of a specific name displays the next initiation of requests belonging to all user IDs.

### ***requestname***

Defines the name of the request for which you want to display information. An asterisk instead of a specific request name displays the next initiations for all requests.

### **FROM *mm/dd/yy***

Specifies the date from which you want to list the run times for the request. The default is the current date.

### **UNTIL *mm/dd/yy***

Specifies the last date for which requests are to appear. If you specify FROM without UNTIL, the UNTIL date defaults to the FROM date, displaying information about requests for that one day only.

### **SYSNAME {\* | *systemname*}**

Specifies the system name of the SSI cluster member for which you want to display request information. If you want the output to only show requests that can run on any system in the SSI cluster, specify \* for SYSNAME. If not specified, the default is to display all requests that meet selection criteria other than the system name.

## Examples

- To list all requests belonging to userid LOUISE whose next run is scheduled to run between June 1 and August 1, enter the following command:  

```
vmsched display louise * (from 06/01/YY until 08/01/YY
```
- To list the next initiation of all requests named REPORT of all users scheduled to run today, enter the following command:  

```
vmsched display * report
```
- To list the next initiation of all requests whose names begin with REP and that belong to all users and whose next initiation falls between May 24 and May 26, enter the following command:  

```
vmsched display * rep* (from 05/24/YY until 05/26/YY
```
- To list the next initiation of all requests belonging to all userids for today, enter the following command:  

```
vmsched display * *
```
- To list the next initiation of all requests belonging to all user IDs for today that can run on any SSI member (no SYSNAME assignment), enter the following command:  

```
vmsched display * * (SYSNAME *
```

## END Command

Use the END command to shut down operation of *CA VM:Schedule*.

END

## INITIATE Command

Use the INITIATE command to stop and start requests that initiate on all user IDs, including *CA VM:Schedule*.

This command has the following capabilities:

- It can start either all requests, or only requests that run on the *CA VM:Schedule* service virtual machine.
- It can stop all requests, or all requests except those that run on the *CA VM:Schedule* service virtual machine.

```
INITiate {OFF {ALL | EXCEPT VMD} | ON {ALL | ONLY VMD}}
```

### Definitions

#### OFF

Stops request processing.

#### ON

Starts request processing.

#### ALL

Applies the INITIATE command to all requests.

#### EXCEPT VMD

Applies the INITIATE command to all requests except the requests that run on *CA VM:Schedule*.

#### ONLY VMD

Applies the INITIATE command only to requests that run on *CA VM:Schedule*.

### Examples

- To stop all request processing, enter the following command:  

```
vmsched initiate off all
```

When *CA VM:Schedule* is reinitialized, any requests that *CA VM:Schedule* releases through the *RELEASE* configuration file record are skipped.
- To stop processing for all requests except the requests that run on the *CA VM:Schedule* service virtual machine, enter the following command:  

```
vmsched initiate off except vmd
```

- To start all request processing, enter the following command:  
`vmsched initiate on all`
- To start processing only requests that run on the *CA VM:Schedule* service virtual machine, enter the following command:  
`vmsched initiate on only vmd`

## OPERATOR Command

Use the OPERATOR command to display the Operator Main Menu.

OPERator

## QLOCK Command

To display all locks that are held by all processes in the server, use the QLOCK command.

QLock

### Description

Use the output from QLOCK along with the output from the QPCB command to determine which user ID is blocking a particular request.

If you need to use this command when the *CA VM:Schedule* process on your machine has not completed, issue it through the CP SMSG command. For example:

```
#cp msg vmsched qlck
```

To determine which user ID is preventing a nonforced shutdown of *CA VM:Schedule*, issue the QLOCK command from the *CA VM:Schedule* service virtual machine.

## QPCB Command

To list currently active processes and the commands they are executing, use the QPCB command.

QPcb

### Description

The QPCB command lists all active processes and the commands they are executing. The command generates a single line of output for each process. Use this command to determine who is using the system before shutting down *CA VM:Schedule* with the END command.

## QREPLY Command

To list any messages that are waiting for a reply, use the QREPLY command.

QREPLY

To reply to any outstanding messages, use the REPLY command.

### Example

To list all the CA VM:Schedule messages that are waiting for responses, and to list the ID numbers for the replies, enter the following command:

```
vmsched qreply
```

A response is given as follows:

ReplyId	Message
1	VMDDAT0104W CPU clock is 17 hours AHEAD OF VMSCHED CLOCK file. Use the command 'REPLY nnn OK' to continue.

Valid replies: OK

## REPLY Command

To reply to a *CA VM:Schedule* message that is waiting for a reply, use the REPLY command.

Reply *number text*

### Definitions

***number***

Defines the number of the message to which you are replying.

***text***

Defines the reply text.

### Examples

- To respond OK to message 01, enter the following command:  
`vmsched reply 01 ok`

## RESET Command

To clear the counter of invalid passwords that a user has entered on CA VM:Schedule commands and screens, use the RESET command.

```
RESET issuer owner
```

### Definitions

***issuer***

Defines the user ID that entered the invalid password.

***owner***

Defines the user ID of the virtual machine for which the incorrect password was entered. An asterisk for owner resets the counters for all user IDs for which a user ID entered an incorrect password.

### Description

RESET can clear the counter for invalid entries that a user has made for another user or for all other users on the system. (Invalid password entries that all users make and for all users are automatically cleared when *CA VM:Schedule* initializes.)

**Note:** The maximum number of attempts a user can make to schedule a request that requires a password is specified on the JOURNAL record in the VMSCHED CONFIG file.

### Examples

- To clear the invalid entries JOHANN has made trying to enter the password for MARCIA, enter:  

```
vmsched reset johann marcia
```
- To clear the invalid entries JOHANN has made trying to enter the password for any user on the system, enter:  

```
vmsched reset johann *
```

## RESUME Command

To restart CA VM:Schedule operation after the SUSPEND command suspends the product operation, use the RESUME command.

RESume

## SUSPEND Command

To stop *CA VM:Schedule* operation temporarily without shutting it down, use the SUSPEND command. Use this feature to:

- Run system backups
- Run the VMDRPT utility

SUSPEND

To resume the system operation of *CA VM:Schedule*, use the RESUME command.

## TRANSFER Command

To change the user ID on which a request runs, use the TRANSFER command. For example, use this command to reassign requests that belong to an employee who left the company or changed departments.

```
TRANSfer {requestname | *} olduser newuser
```

### Definitions

***requestname***

Defines the name of the request to be transferred. An asterisk instead of a specific request name specifies all requests that currently belong to *olduser*.

***olduser***

Defines the user ID that currently owns the request.

***newuser***

Defines the user ID to which the request is transferred.

### Restrictions

*CA VM:Schedule* prevents duplicate request names. You therefore cannot transfer a request to another user who has a request of the same name pending. Attempts to transfer requests that result in duplicate names fail. *CA VM:Schedule* rejects the transfer with an error message and a CMS return code of 24.

You must have TRANSFER authorization to use this command.

### Examples

- To transfer all the requests that userid HARRY owns to userid DALE, enter the following command:  

```
vmsched transfer * harry dale
```
- To transfer the ACCOUNTS request from userid ALICE to userid LOUISE, enter the following command:  

```
vmsched transfer accounts alice louise
```

## WHEN Command

Use the WHEN command to list times when a request runs.

If you do not specify any options, CA VM:Schedule displays up to 999 initiations that are scheduled for today.

```
WHEN requestname [ ("Options")]
```

Options:

```
[FRom mm/dd/yy]  
[NExt number]  
[UNtil mm/dd/yy]  
[File filename]  
[PRint]  
[STack {FIFO | LIFO}]  
[Term]  
[Password password]  
[SOrt]  
[User userid]
```

## Definitions

### ***requestname***

Specifies the name of a request for which you want to list the run times. An asterisk displays all request names; a trailing asterisk lists requests with similar names.

### **FROM *mm/dd/yy***

Specifies the date from which you want to list the run times for the request. The default is the current date.

### **UNTIL *mm/dd/yy***

Specifies the last date for which the WHEN command displays requests. If FROM is specified without UNTIL, the UNTIL date defaults to the FROM date.

**NEXT *number***

Specifies how many initiations to list. The default is 999. Any date range that you specify in addition to a NEXT parameter limits the number of requests that the WHEN command displays. See *Controlling the Number of Runs Displayed* (see page 58) for information about using this option.

**FILE *filename***

Writes the output of the WHEN command to the file *filename* VMSCHED A1. The default output destination is your terminal.

**PRINT**

Sends the output from the WHEN command to your virtual printer. The default output destination is your terminal.

**STACK { FIFO | LIFO }**

Specifies to stack output in your program stack first in, first out (FIFO) or last in, first out (LIFO).

**TERM**

Displays the output on your terminal. This option is the default if no other output option is specified.

**PASSWORD *password***

Specifies the CP LOGON password for the user ID whose requests are being queried. The password must be specified when you are listing run times for the requests that another user ID owns. Your *CA VM:Schedule* system administrator can specify that users are not allowed to enter the password in the option string. In this case, you are required to use the prompting procedure for password entry. You do not have to supply your password when listing times for requests you scheduled on your own user ID.

**USER *userid***

Specifies the user ID for whose requests the WHEN command is to list run times. The default is the user ID that issued the WHEN command. If you specify another user ID, the CP logon password must be specified unless you have the NOPASS authorization.

**SORT**

Sorts the list of scheduled requests by date and time. This option is useful when using pattern matching to display multiple request names. By default, repeat initiations within a given request name are already shown in order of the date and time.

## Output Destinations

You can include the FILE, PRINT, STACK, and TERM options together on a single WHEN command line. However, when you enter the WHEN command from the *CA VM:Schedule* console, you can include only the TERM option. If you do not specify any output option (FILE, PRINT, STACK, or TERM), *CA VM:Schedule* displays the output on your terminal.

## Controlling the Number of Runs Displayed

If you do not include the NEXT, FROM, or UNTIL options, *CA VM:Schedule* lists the times and dates for the next 999 runs starting today. If you specify dates only, *CA VM:Schedule* lists up to 999 runs for the date range specified.

## Listing Many Requests

To avoid tying up your terminal while listing many requests, perform the following actions:

- Schedule the WHEN command run for a later time, when you do not need your user ID to perform interactive work.
- Send the output to a printer.

## Examples

- To display all requests that are scheduled to run on your userid from September 7, through September 11, enter the following command:  

```
vmsched when * (from 09/07/YY until 09/11/YY
```
- You scheduled the CHECKUP request to run every 15 minutes today. To ensure that the request was scheduled correctly, display the next ten runs by entering the following command:  

```
vmsched when checkup (next 10
```
- The system is scheduled to be down for the maintenance over the weekend of 08/06/YY through 08/08/YY. To list requests for the user DIANE scheduled to run over this weekend, and sort them by date and time, enter the following command:  

```
vmsched when * (user diane from 08/06/YY until 08/08/YY sort
```

- To list the next ten runs of the REPORT request that are scheduled to run today, enter the following command:

```
vmsched when report (next 10
```

- To display all requests that are scheduled to run on November 28 and 29, perform the following action. Place the following command in an EXEC and schedule the EXEC to run after you have logged off for the day:

```
vmsched when * (from 11/28/YY until 11/29/YY user * print sort
```



# Index

---

## A

- access VM • 18
  - Schedule • 18
- autolog • 17
  - definition • 17

## C

- CMS HELP command • 19
- commands, general • 19, 31, 35
  - CMS subset • 31
  - help with • 19
  - quick reference for operators • 35

## D

- disconnected mode • 17

## E

- exit VM • 18
  - Schedule • 18

## H

- HELP command • 19
  - CMS • 19
- HELP command (CMS) • 20
  - help for system messages • 20

## L

- leave VM • 18
  - Schedule • 18
- listing • 22, 24, 30, 32
  - class status • 30
  - messages waiting for replies • 32
  - other users' requests • 22, 24
  - request characteristics • 22

## M

- menus • 19
  - CMS HELP • 19
- messages • 19, 20, 32
  - answering VM • 32
    - Schedule system • 32
  - help for • 20
  - help with • 19

- system, help for • 20

## O

- online help • 19

## Q

- quick reference • 35
  - operator commands • 35

## R

- requests • 17, 22, 26, 30, 56
  - definition • 17
  - listing other users' • 22, 56
  - listing run times • 56
  - managing classes • 30
  - starting • 26
  - stopping • 26

## S

- scheduled requests • 17
  - definition • 17
- screens • 23, 24, 26, 28, 30
  - Control Request Classes • 30
  - Display Future Initiation(s) of a Request • 24
  - Display Pending Initiations • 23
  - Initiations of Scheduled Requests • 26
  - Reset a Password Count to Zero • 28
  - WHEN command output • 24
- start VM • 18
  - Schedule • 18
- system messages • 19
  - help with • 19