Guide to Installing and Using WATCHER

June, 1994

This manual describes the installation, configuration, and operation of WATCHER, an idle terminal monitor for VMS systems.

Revision/Update Information:	This is a revised manual.
Operating System and Version:	VAX/VMS V5.0 or later, OpenVMS AXP V1.0 or later
Software Version:	WATCHER V2.9

Matthew D. Madison
MadGoat Software

24 June 1994

Permission is granted to copy and redistribute this document for no commercial gain.

The information in this document is subject to change without notice and should not be construed as a commitment by the author. The author assumes no responsibility for any errors that may appear in this document.

DISCLAIMER: The author, the author's employer, and MadGoat Software make no representations or warranties with respect to the contents hereof and specifically disclaim any implied warranties of merchantability or fitness for any particular purpose.

The following are trademarks of Digital Equipment Corporation:

Copyright ©1993, 1994 MadGoat Software. All rights reserved.

Contents

PREFACE v

CHAPTER 1 INSTALLING WATCHER 1–1

1.1 REQUIRED LOGICAL NAMES 1–1
1.1.1 Logical Name for Help Library 1–1

1.2 PRIVILEGES REQUIRED 1–2

1.3 OTHER REQUIREMENTS 1–2

CHAPTER 2 CONFIGURING WATCHER 2–1

2.1 SETTING UP WATCH RULES 2–1
2.1.1 Identifying Terminals 2–1
2.1.2 WATCH Criteria 2–2
2.1.3 Terminal Groupings 2–2

2.2 EXCLUSIONS AND OVERRIDES 2–2

2.3 SAVING CONFIGURATIONS 2–3
2.3.1 VAXcluster Environments 2–3
2.3.2 Editing Configurations 2–3

2.4 DECWINDOWS SUPPORT 2–4

CHAPTER 3 TROUBLESHOOTING WATCHER 3–1

3.1 FORCING WAKEUPS 3–1

PART II COMMAND DESCRIPTIONS

WCP	CMD–3
@ (REDIRECT COMMAND INPUT)	CMD–4
EXCLUDE	CMD–5
EXIT	CMD–7
HELP	CMD–8
OVERRIDE	CMD–9
QUIT	CMD–12
RESET	CMD–13
SAVE	CMD–14
SET ACTION	CMD–15
SET BELL	CMD–16
SET DAYS	CMD–17
SET DEBUG	CMD–18
SET DECWINDOWS	CMD–19
SET EVENT_LOG	CMD–20
SET INSWAP	CMD–21
SET INTERVAL	CMD–22
SET MULTIWARN	CMD–23
SET VERIFY	CMD–24
SET WATCH_DEFAULT	CMD–25
SHOW	CMD–27
SHUTDOWN	CMD–28
WATCH	CMD–29

Preface

One of the first programs a new VMS system manager usually needs is an ‘‘idle terminal monitor’’ (ITM). That is, a program to monitor terminal activity and logout those users whose terminals remain inactive for an extended period of time. An ITM helps ensure that system resources are not wasted and helps reduce the possibility of intruders using unattended terminals as a means of entry into the system.

Unfortunately, an ITM can also be an annoyance to system users. A simple ITM can victimize legitimate users who may need to remain logged in but idle while they are at work. This can lead to clever users devising ‘‘hacks’’ to evade the ITM, defeating the purpose of using the ITM in the first place.

WATCHER has a high degree of flexibility, allowing system managers to decide how to accommodate users' needs while still addressing operational and security issues. WATCHER is fully configurable, providing the following features:

- You can tell WATCHER which terminals to watch, and on a per-terminal basis, what measurements (CPU use, process I/O count, terminal I/O count) to use as criteria for determining idleness, and how long a terminal should be idle before the user should be forced off.
- Users can be excluded from interference by WATCHER based on any combination of username, UIC, a held identifier, privileges, terminal device and/or port name, time-of-day/day-of-week, and name of image being run.
- You can override or modify the watch criteria and/or idle times for any user based on any combination of username, UIC, a held identifier, privileges, terminal device and/or port name, time-of-day/day-of-week, and name of image being run.

Through the use of these features, the system manager should be able to configure WATCHER to handle most types of terminals and accommodate most users.

Intended Audience

This manual is intended for the system manager or other person responsible for installing and configuring WATCHER.

Document Structure

This document consists of two parts. The first describes the installation and use of WATCHER. The second describes all of the WATCHER Control Program (WCP) commands in detail.

Support for WATCHER

There is no formal support for WATCHER. If you have Internet connectivity, however, you may wish to subscribe to one or more of the following MadGoat Software mailing lists:

Info-MadGoat@wkuvx1.wku.edu

Discussion of MadGoat Software products by users and MadGoat developers. To subscribe, send a message to *Info-MadGoat-Request@wkuvx1.wku.edu* with the word SUBSCRIBE in the first line of the body of the message.

MadGoat-Announce@wkuvx1.wku.edu

Announcements of new releases and new products from MadGoat. To subscribe, send a message to *MadGoat-Announce-Request@wkuvx1.wku.edu* with the word SUBSCRIBE in the first line of the body of the message.

MadGoat-Bugs@wkuvx1.wku.edu

Address for reporting bugs in MadGoat Software products. Please include the name of the package and version in the subject header of the message, so the report can be more easily directed to the appropriate developer.

If you cannot send electronic mail, you can contact the author by post, facsimile, or telephone at:

Matthew Madison
TGV, Incorporated
101 Cooper Street
Santa Cruz, CA 95060 USA

Fax: +1 408 457 5208
Phone: +1 408 457 5390

1 Installing WATCHER

To use WATCHER, you need the following files:

WATCHER.EXE	The main WATCHER image
WCP.EXE	The WATCHER Control Program
WCP_HELPLIB.HLB	Help library for WCP
DECW_STARTLOGIN.COM	Part of DECwindows support
WATCHER_CONFIG.WCFG	You create this file with WCP
WATCHER_STARTUP.COM	Sample startup command procedure
WATCHER_SHUTDOWN.COM	Sample shutdown command procedure
SAMPLE_CONFIG.WCP	Sample configuration commands

The package comes with the object code files and libraries and a command procedure called LINK.COM, for creating the two images.

It is easiest to simply place all of the files in the distribution in one directory, run LINK.COM to create the images, then edit WATCHER_STARTUP.COM and the sample configuration commands in SAMPLE_CONFIG.WCP as needed for your system. Then all you need to do is to run WCP, execute the WCP command file you created from the sample, which in turn creates a WATCHER_CONFIG.WCFG file, then execute WATCHER_STARTUP.COM to start the Watcher process.

1.1 Required Logical Names

The three system-wide logical names WATCHER requires are:

WATCHER_DIR	Should point to location of images and command procedures
WATCHER_CONFIG	Configuration file to be used
WATCHER_TRACE	Trace file; use NL: if debug disabled

They should all be defined in executive mode.

1.1.1 Logical Name for Help Library

The help library for WCP may be placed in SYS$HELP, or, if you define the logical name WCP_HELPLIB to be the full path name of the file, anywhere else on the system. The sample WATCHER_STARTUP.COM includes the necessary DEFINE command to do this for you.

1.2 Privileges Required

The account that is used for the WATCHER process requires the following privileges:

CMKRNL	Required for DECwindows support and disconnects
PRMMBX	For defining the command mailbox
PSWAPM	Required for disconnects
SHARE	For sending warning messages to other users' terminals
SYSNAM	For defining the command mailbox
SYSPRV	(optional) to ensure access to appropriate files
WORLD	For getting information about and killing processes

SYSPRV is not needed if you make sure that WATCHER has enough access to read its configuration files and the system rightslist (if using identifiers as an exclusion criterion on pre-VMS V5.4 systems), and write its log and trace files (if used). Both CMKRNL and SYSPRV are required for DECwindows support. CMKRNL and PSWAPM are required to perform virtual terminal disconnections.

1.3 Other Requirements

The RUN command in WATCHER_STARTUP.COM should provide the WATCHER process with sufficient quotas to operate on most systems. CPU and memory requirements will vary depending on the number of rules in the WATCHER configuration, peak number of interactive users, and peak number of watched users. You may wish to refer to the following table in computing expected memory resources needed by the WATCHER process:

Memory required per WATCH rule	206 bytes
Memory required per EXCLUDE or OVERRIDE rule	507 bytes
Memory required per interactive process	465 bytes
Memory required per watched process	531 bytes
Size of WATCHER code (approximate)	18K bytes

On pre-VMS V5.4 systems, allow a small increase in CPU, memory, and I/O requirements if identifiers are used as an exclusion mechanism, since WATCHER will require access to the rightslist database for each interactive process. DECwindows support also requires additional overhead for access to the job logical name tables of all interactive and detached processes on the system.

2 Configuring WATCHER

The WATCHER Control Program (WCP) is used to create WATCHER configurations. WCP is designed to be executed as a VMS foreign command. To set up the foreign command, define the symbol

```
$ WCP :== $WATCHER_DIR:WCP
```

Once the symbol is set up, you can invoke WCP with the command:

```
$ WCP
```

WCP will automatically load the contents of your defined WATCHER_ CONFIG file, if it exists.

2.1 Setting up WATCH Rules

The WATCH command sets up rules that determine which terminals get watched, how to determine whether the terminals are active, and how long terminals must be inactive before a user can be forced off. For example:

```
WCP> WATCH *$RT*/MEASURE=PROCESS_IO/LOGOUT=00:15:00
```

This command sets up a rule for watching all DECnet remote logins, using changes in total process I/O (buffered plus direct) to determine process activity, and causing logouts to occur after 15 minutes of inactivity.

Note: You must have at least one WATCH command in your configuration.

2.1.1 Identifying Terminals

WATCH commands take any wildcard pattern. All terminal device names that match the specified pattern are watched. The device names used by WATCHER are the physical device names of terminals; if the system is part of a VAXcluster, SCS node name is prefixed to the device name, as is normally done by VMS with cluster-accessible devices.

If the terminal device driver supports remote port identification, as does the LTDRIVER for LAT terminals, the remote port information can also be used as a match criterion by using the /ACCPORNAM qualifier. The port name can be specified as a wildcard pattern. For example:

```
WCP> WATCH *$LT*/ACCPORNAM="TRMSRV/*"
```

This command would cause the terminals attached to terminal server TRMSRV to be watched.

2.1.2 WATCH Criteria

WATCHER gives you the choice of using one or more of the following measurements as criteria for judging whether a terminal or user is active:

TERMINAL_IO	the I/O operation count on the terminal device
CPU	The total CPU time used by the process owning the terminal plus all of its subprocesses, in centiseconds
PROCESS_IO	the sum of the buffered and direct I/O counts of the process owning the terminal plus all of its subprocesses

The TERMINAL_IO measurement is useful for conventional terminals but cannot be used for workstations (running either VWS or DECwindows) due to the nature of workstation activity. PROCESS_IO is recommended for use on workstation terminal devices.

For any of these measurements you can specify a minimum threshold value. When WATCHER performs a comparison, the difference between the current measured value and the last measured value must be greater than the specified threshold to be counted as activity. The default threshold value is zero, so that any difference at all counts as activity.

Several samples of WATCH commands with different criteria and threshold values are provided in SAMPLE_CONFIG.WCP.

2.1.3 Terminal Groupings

You can group WATCH rules together by using the /GROUP qualifier. When WATCHER applies its rules for determining terminal activity, activity on one terminal in the group counts as activity for all the terminals in the group. The main use for this feature is with multi-windowed terminals and workstations running VWS (DECwindows workstations are handled in this manner automatically). For example, the following rules handle all the workstation terminal types on a standalone VWS workstation:

```
WCP> WATCH WTA*/GROUP=VWS   ! normal VT200-series windows
WCP> WATCH TKA*/GROUP=VWS   ! Tek 4010 emulation windows
WCP> WATCH TJA*/GROUP=VWS   ! Tek 4125 emulation windows
```

The user can then create any number of any type of terminal window, and as long as one of them is active, they will all remain logged in.

2.2 Exclusions and Overrides

WATCHER's behaviour towards a terminal or user can be modified through the definition of exclusion and override rules. Exclusions and overrides can be based on any combination of username, terminal/port name, UIC, image being run, privileges, a held identifier, and time of day. Exclusion rules prevent WATCHER from taking any action towards a user, while override rules merely modify how the terminal is watched (i.e., the activity criteria and inactivity periods). For example (taken from a VAXcluster system):

```
WCP> EXCLUDE SYSTEM/TERMINAL=*$OPA0:
WCP> OVERRIDE JONES/TERMINAL=NODE1$TXA3:/DURING=(PRIMARY:8-16)-
_WCP>           /LOGOUT=02:00:00
```

The first command prevents WATCHER from taking any action against the SYSTEM account while it is logged into the system console. The second command extends the logout inactivity period to two hours for user JONES weekdays from 8 am to 4:59 pm, while JONES is logged into the terminal in her office, which is on port TXA3 on system NODE1.

2.3 Saving Configurations

Once you have established the rules you need for your configuration, you should create the configuration file with the SAVE command:

```
WCP> SAVE WATCHER_CONFIG
```

If WATCHER is currently running, you can have the new configuration take effect immediately with the RESET command, which will cause the WATCHER process to reload its configuration information from the file.

2.3.1 VAXcluster Environments

For mainly homogeneous VAXcluster environments, you should be able to use one configuration file for all nodes in the cluster. If you have a mix of nodes, however, it may be easier to create multiple configuration files and define the WATCHER_CONFIG logical name differently depending on the system.

2.3.2 Editing Configurations

The WATCH, EXCLUDE, and OVERRIDE commands all have a /DELETE to allow you to remove rules from the database, and you can add rules as well. However, you cannot control the order of the new rules (order is important because WATCHER searches the rule lists in the order you enter them until one matches). To assist in making complex changes to the configuration, the SHOW command has a /COMMAND qualifier that causes the configuration information to be displayed as commands you would enter to build the configuration:

```
WCP> SHOW/COMMAND/OUTPUT=CONFIG.WCP ALL
```

Once you dump the commands to the command file, you can edit the command file as needed and create a new configuration with the commands:

```
$ WCP/NOFILE
WCP> @CONFIG
WCP> SAVE WATCHER_CONFIG
```

Instead of editing the configuration, it may be easier just to maintain a WCP command file with the necessary commands in it and build a new configuration each time you need to make a change.

2.4 DECwindows Support

The VMS DECwindows implementation makes it difficult for a WATCHER-type program to properly identify, warn, and logout DECwindows sessions. However, WATCHER does provide limited support for watching DECwindows sessions, enabled with the following commands:

```
WCP> SET DECWINDOWS
WCP> WATCH *WSA*/MEASURE=PROCESS_IO/NOWARNING
```

Note that you cannot use TERMINAL_IO as a measurement when watching DECwindows sessions, nor can WATCHER give warnings to idle DECwindows sessions.

WATCHER identifies DECwindows sessions by searching the job logical name table for each interactive process for the logical name DECW$DISPLAY, defined in executive mode. Each interactive job related to a single DECwindows session will have the same value for DECW$DISPLAY. WATCHER immediately changes the terminal device name it uses to the WSA device name (even for DECterm sessions) and also sets the group name to the WSA device name. In this way, activity in any of the DECwindows jobs will be counted as activity for all jobs related to that session.

The DECwindows window manager and DECterm controller processes are detached processes that are also needed by WATCHER (when forcing off a DECwindows session). To identify these processes, WATCHER searches for detached processes with DECW$DISPLAY defined in user mode. WATCHER tracks these processes, but does not use them in activity determination (it calls them ‘‘fake’’ processes in debug/trace logs).

When WATCHER identifies a DECwindows session to be forced off, it looks for all processes (including the detached processes, which are important) with a matching WSA device name and forces them off. This should destroy all the windows on the workstation and return it to a blank, background screen. It then creates a detached process that executes the DECW_STARTLOGIN command procedure (which must be located in WATCHER_DIR:), which, after waiting a few seconds for other activity to die down, restarts the login process on the affected WSA device.

WATCHER cannot be used to watch DECwindows jobs that are started on remote systems, with the local workstation being used only as a display. There must be at least some jobs running on the workstation with some activity to prevent WATCHER from logging out the DECwindows session.

This technique should be effective for VMS DECwindows V2 (VMS V5.1 through V5.5) and V3 (also known as DECwindows/Motif V1.0), and should even work with X terminals. It may not work with future DECwindows implementations.

3 Troubleshooting WATCHER

If WATCHER is not behaving as expected, there may be a problem with your WATCHER configuration. There is debug/trace code built into WATCHER to allow you to monitor five categories of activities: the mainline WATCHER code, the exclusion-checking code, the override-checking code, measurement checks, and process information collection. Through the use of the SET DEBUG command, you can turn on tracing for any or all of these debugging categories.

If WATCHER is already running, the best way to set up a test configuration is with the following command sequence:

```
$ SET PROCESS/PRIVILEGE=(SYSNAM,SYSPRV)
$ WCP
WCP> SET DEBUG=n
WCP> SET NOACTION
WCP> SAVE WATCHER_DIR:TEST_CONFIG
WCP> EXIT
$ DEFINE/SYSTEM/EXEC WATCHER_CONFIG WATCHER_DIR:TEST_CONFIG
$ DEFINE/SYSTEM/EXEC WATCHER_TRACE  trace-file-spec
$ WCP RESET
```

The debug level *n* is described in the SET DEBUG command description, but usually will be 1 (just mainline) or 31 (full). You can direct the trace information to any file accessible to WATCHER, or to an unowned terminal.

The SET NOACTION command will prevent WATCHER from actually logging anyone out or sending warning messages to terminals.

Subsequent WCP RESET commands will cause the trace file to be closed and a new version created, so you can easily view past trace information. To go back to ‘‘production’’ mode, just redefine WATCHER_CONFIG back to the name of the real configuration file, define WATCHER_TRACE to be NL:, and issue another WCP RESET command.

3.1 Forcing Wakeups

To assist in debugging, you may want to have WATCHER wake up more often than normal. You can do this by setting a shorter wakeup interval in the test configuration, or you can force a wakeup to occur by writing to the WATCHER control mailbox. From a suitably privileged account (SYSPRV), you can use the commands:

```
$ OPEN/WRITE WMBOX WATCHER_MBOX:
$ WRITE WMBOX ""
$ CLOSE WMBOX
```

Each WRITE command will trigger a wakeup, and WATCHER will go through its processing sequence.

Part II Command Descriptions

WCP

Executes the WATCHER Control Program.

FORMAT

WCP *[command]*

Command Qualifiers	Defaults
/FILE=file-spec	*See description.*

PARAMETERS

[command]

Any WCP command except the input redirection operator (@). The specified command is executed and control is returned to DCL immediately thereafter.

DESCRIPTION

WCP is intended to be used as a DCL ''foreign'' command. To use it as a foreign command, you must define a symbol as follows:

```
$ WCP :== $WATCHER_EXE:WCP
```

Defining the symbol in this way allows you to use the /FILE qualifier and specify ''one-shot'' commands on the command line.

QUALIFIERS

/[NO]FILE=file-spec

Loads the specified WATCHER configuration file for editing. If not specified, the configuration file pointed to by the logical name WATCHER_ CONFIG is loaded. The default file type is WCFG. If /NOFILE is specified, no configuration file is loaded.

@ (Redirect Command Input)

Executes WCP commands read from a file.

FORMAT

@ *file-spec*

PARAMETERS

file-spec

Name of the file containing WCP commands. If omitted, the default file type is WCP.

DESCRIPTION

Use this command to have WCP take further command input from the specified file. There is no built-in limit on the number of levels of nesting of command files, so be careful when using input redirection from within a command file. Commands read from command files are not displayed unless you SET VERIFY.

Command redirection can only be used at the WCP command prompt, not as a ''one-shot'' WCP command. To have a file be used for input for an entire WCP session, use the following sequence of DCL commands.

```
$ DEFINE/USER SYS$INPUT file-spec
$ WCP
```

EXCLUDE

Defines an exclusion rule.

FORMAT

EXCLUDE *username-pat*

Command Qualifiers	Defaults
/ACCPORNAM=port-pat	*/ACCPORNAM=**
/DELETE	
/DURING=daytim-list	*(all the time)*
/HOLDING=identifier	*(ignored)*
/IMAGE=fspec-pat	*/IMAGE=**
/PRIVILEGES=priv-list	*(ignored)*
/TERMINAL=dev-pat	*/TERMINAL=**
/UIC=uic	*/UIC=[*,*]*

PARAMETERS

username-pat

A VMS username or pattern containing wildcards, identifying the user to be excluded.

DESCRIPTION

This command is used to add or remove (with /DELETE) an exclusion rule to the WATCHER configuration. When WATCHER is running, any process that matches all of the specified criteria is not watched. Omitted criteria are not used or always match.

QUALIFIERS

/ACCPORNAM=port-pat

Port name or pattern containing wildcards, identifying the terminal port (for terminal servers and other devices using port names) on which the user must be logged in to be excluded. The default is any port.

/DELETE

Specifies that the rule should be deleted from the configuration. All criteria must match exactly for the rule to be deleted.

/DURING=daytim-list

Specifies a list of days and times during which the user is to be excluded from watching. The day/time specifications are of the form

day:(hour-range[,...])

where *day* is a day of the week or the word PRIMARY or SECONDARY, identifying the primary and secondary days set with SET DAYS, and *hour-range* is either a single hour number (0 through 23) or two hour numbers separated by a hyphen. Multiple hour ranges may be specified per day.

/HOLDING=identifier

Specifies that the user should be excluded if holding the specified identifier. The identifier is converted to binary format before being stored in the configuration, so you must create the configuration file on the target system, or on a node with the same RIGHTSLIST database as the target system, to prevent misinterpretation of the identifier.

If you are running VMS V5.4 or later, WATCHER obtains the identifiers held by the process directly (using $GETJPI), and thus can check identifiers that are granted dynamically. Prior to V5.4, WATCHER uses the $FIND_HELD system service to scan the system rightslist for identifiers held by the user that owns each process; dynamically granted identifiers cannot be checked in this case.

Note that if you do not use /HOLDING on any EXCLUDE or OVERRIDE command, the rights identifier information is not collected by WATCHER, resulting in some savings in processing time (especially on pre-V5.4 systems).

/IMAGE=fspec-pat

Specifies that the user should be excluded if running an executable image whose name (as returned by the JPI$_IMAGNAME item from $GETJPI) matches the specified wildcard pattern. For a job with subprocesses, the image name that WATCHER uses for matching against the wildcard pattern is the image currently being run by the master process in the job, or, if the master process is not running an image, an image being run by one of the subprocesses (randomly selected if there are two or more such subuprocesses).

/PRIVILEGES=priv-list

Specifies that the user is to be excluded only when holding the specified privilege or privileges. If omitted, privileges are not used as a criterion.

/TERMINAL=dev-pat

Specifies a terminal device name or pattern containing wildcards. The user must be logged into a matching terminal to be excluded.

/UIC=uic

Specifies a UIC or UIC pattern. A process is excluded only when owned by a matching UIC. You may use an asterisk for the member part of the UIC to have all UIC's in a group match. If omitted, any UIC will match.

EXIT

Ends a WCP session.

FORMAT EXIT

DESCRIPTION

Ends the current WCP session and returns control to DCL. If you have modified the configuration, EXIT will ask for a file name for saving the configuration before exiting.

HELP

Displays help information.

FORMAT

HELP *[topic...]*

PARAMETERS

topic

The name of a topic in the help library. If omitted, a list of topics is displayed.

DESCRIPTION

This command is pretty straightforward.

OVERRIDE

Defines an override rule.

FORMAT

OVERRIDE *username-pat*

Command Qualifiers	Defaults
/ACCPORNAM=port-pat	*/ACCPORNAM=**
/DELETE	
/DURING=daytim-list	*(all the time)*
/HOLDING=identifier	*(ignored)*
/IMAGE=fspec-pat	*/IMAGE=**
/PRIVILEGES=priv-list	*(ignored)*
/TERMINAL=dev-pat	*/TERMINAL=**
/UIC=uic	*/UIC=[*,*]*
/[NO]DISCONNECT[=deltatime]	
/[NO]FORCE_EXIT[=deltatime]	
/[NO]LOGOUT[=deltatime]	
/MEASURE=(measurement[,...])	
/[NO]WARNING[=deltatime]	

PARAMETERS

username-pat

A VMS username or pattern containing wildcards, identifying the user for which the override is to take effect.

DESCRIPTION

This command is used to add or remove (with /DELETE) an override rule to the WATCHER configuration. When WATCHER is running, any process that matches all of the specified criteria will have the warning, logout, and measurement information, if specified, taken from the override rule instead of the WATCH rule.

QUALIFIERS

/ACCPORNAM=port-pat

Port name or pattern containing wildcards, identifying the terminal port (for terminal servers and other devices using port names) on which the user must be logged in to have the override apply. The default is any port.

/DELETE

Specifies that the rule should be deleted from the configuration. All criteria must match exactly for the rule to be deleted.

/DURING=daytim-list

Specifies a list of days and times during which the override is to apply. The day/time specifications are of the form

day:(hour-range[,...])

where *day* is a day of the week or the word PRIMARY or SECONDARY, identifying the primary and secondary days set with SET DAYS, and *hour-range* is either a single hour number (0 through 23) or two hour numbers separated by a hyphen. Multiple hour ranges may be specified per day.

/HOLDING=identifier

Specifies that the override should apply only if the user is holding the specified identifier. The identifier is converted to binary format before being stored in the configuration, so you must create the configuration file on the target system, or on a node with the same RIGHTSLIST database as the target system, to prevent misinterpretation of the identifier.

If you are running VMS V5.4 or later, WATCHER obtains the identifiers held by the process directly (using $GETJPI), and thus can check identifiers that are granted dynamically. Prior to V5.4, WATCHER uses the $FIND_HELD system service to scan the system rightslist for identifiers held by the user that owns each process; dynamically granted identifiers cannot be checked in this case.

Note that if you do not use /HOLDING on any EXCLUDE or OVERRIDE command, the rights identifier information is not collected by WATCHER, resulting in some savings in processing time (especially on pre-VMS V5.4 systems).

/IMAGE=fspec-pat

Specifies that the override should apply only if the user is running an executable image whose name (as returned by the JPI$_IMAGNAME item from $GETJPI) matches the specified wildcard pattern. For a job with subprocesses, the image name that WATCHER uses for matching against the wildcard pattern is the image currently being run by the master process in the job, or, if the master process is not running an image, an image being run by one of the subprocesses (randomly selected if there are two or more such subuprocesses).

/PRIVILEGES=priv-list

Specifies that the override should apply only when the user is holding the specified privilege or privileges. If omitted, privileges are not used as a criterion.

/TERMINAL=dev-pat

Specifies a terminal device name or pattern containing wildcards. The user must be logged into a matching terminal to have the override apply.

/UIC=uic

Specifies a UIC or UIC pattern. The override applies only to processes owned by a matching UIC. You may use an asterisk for the member part of the UIC to have all UIC's in a group match. If omitted, any UIC will match.

/[NO]DISCONNECT[=deltatime]

Specifies that the logout/disconnect information should be overridden, performing a virtual terminal disconnection instead of deleting the user process. The inactivity interval can be overridden by specifying a *deltatime*, or logouts/disconnects can be prevented altogether by specifying /NODISCONNECT (although it is more efficient to use EXCLUDE for this).

/[NO]FORCE_EXIT[=deltatime]

Specifies that the logout/disconnect information should be overridden, performing a forced image exit instead of deleting the process or disconnecting the terminal. Only user-mode images are forced; if the user is at DCL command level, the forced exit is skipped. The inactivity interval can be overridden by specifying a *deltatime*, or exits/logouts/disconnects can be prevented altogether by specifying /NOFORCE_EXIT (although it is more efficient to use EXCLUDE for this).

/[NO]LOGOUT[=deltatime]

Specifies that the logout/disconnect information should be overridden, performing a process deletion instead of a virtual terminal disconnection. The inactivity interval can be overridden by specifying a *deltatime*, or logouts can be prevented by specifying /NOLOGOUT (although it is more efficient to use EXCLUDE for this).

/MEASURE=(measurement[,...])

Specifies that the activity measurements should be overridden. For *measurement*, specify one of the following:

CPU[:*threshold*]	CPU time (the sum of the CPU time used by the process and all its subprocesses, in centiseconds) should be used as a criterion. If *threshold* is specified, the difference in CPU time between passes must exceed the specified threshold for a process to be considered active.
PROCESS_IO[:*threshold*]	Process I/O (the sum of the buffered and direct I/O counts for the process and all its subprocesses) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.
TERMINAL_IO[:*threshold*]	Terminal I/O (the operation count on the terminal device) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.

Any combination of PROCESS_IO, CPU, and TERMINAL_IO is permitted. If omitted, *threshold* values default to zero. Note that threshold values should be chosen as a function of the wakeup interval (defined with SET INTERVAL).

/[NO]WARNING[=deltatime]

Specifies that the warning information should be overridden. The warning inactivity interval can be overridden by specifying a *deltatime*, or warnings can be prevented by specifying /NOWARNING.

QUIT

Quits WCP without saving configuration changes.

FORMAT QUIT

DESCRIPTION

If changes to the configuration have been made, you are asked for confirmation before quitting.

RESET

Sends a reset command to the WATCHER process.

FORMAT RESET

DESCRIPTION

This command sends a reset command to the WATCHER process, which causes WATCHER to flush all process and configuration information, close its log and trace files, and read in the configuration again.

OPER and SYSPRV privileges are required for this command.

SAVE

Saves a WATCHER configuration.

FORMAT SAVE *[file-spec]*

PARAMETERS *file-spec*

Name of the file to which the configuration should be written. If omitted, it defaults to the name of the file read in with the WCP/FILE qualifier (if any). If specified, the default file type is WCFG and the default location is the current default directory.

SET ACTION

Controls whether WATCHER performs warning and logout actions.

FORMAT SET [NO]ACTION

DESCRIPTION

To test out a WATCHER installation or configuration change, you may want to SET NOACTION to prevent WATCHER from actually notifying any terminals or logging any users out. The SET NOACTION setting allows you to test your configuration safely, and use the DEBUG and trace facilities to see how WATCHER would have performed.

When you are through testing, return the setting back to the default, SET ACTION, to have WATCHER actually perform warnings, disconnections, logouts, etc.

SET BELL

Enables or disables the ringing of the terminal bell on warnings and logouts.

FORMAT SET [NO]BELL

DESCRIPTION

By default, the terminal bell is not rung when WATCHER displays a warning or logout message on a terminal. SET BELL will cause WATCHER to send a BEL character with the message to cause the terminal bell to ring.

SET DAYS

Establishes the primary and secondary day settings for subsequent commands.

FORMAT

SET DAYS

Command Qualifiers	Defaults
/PRIMARY=(day-list)	
/SECONDARY=(day-list)	

DESCRIPTION

This command is used to move one or more days from the primary day list to the secondary day list or vice-versa. These lists are used as shorthand by other commands when you specify PRIMARY or SECONDARY on a /DURING qualifier.

SET DEBUG

Enables or disables debug tracing and sets the level of debug information.

FORMAT SET [NO]DEBUG[=mask]

DESCRIPTION

SET DEBUG turns on debug tracing. WATCHER will send debug output to the file or device identified by the WATCHER_TRACE logical name. For *mask*, specify a decimal number representing a bitmask indicating which kinds of debugging information you want logged.

Value	Bit no.	Description
1	0	main line code
2	1	exclusion checks
4	2	override checks
8	3	measurement checks
16	4	process info collection

The *mask* value can be any one of the above, or a sum of any of the above values. Use SET NODEBUG to disable debug tracing.

SET DECWINDOWS

Enables or disables extra processing required for DECwindows support.

FORMAT

SET [NO]DECWINDOWS

DESCRIPTION

Since the DECwindows support requires additional processing beyond that used for watching normal terminals, it is by default turned off. You should only SET DECWINDOWS when you will be running WATCHER on a DECwindows workstation (or a system with DECwindows terminals). You will also need to include a WATCH rule for WSA terminal devices to watch DECwindows sessions (as described in Section 2.4).

SET EVENT_LOG

Establishes how normal WATCHER events are recorded.

FORMAT

SET [NO]EVENT_LOG

Command Qualifiers	Defaults
/FILE=file-spec	
/OPERATOR=oper-list	

DESCRIPTION

This command specifies how normal WATCHER events (startup, shutdown, reset, and logout events) are recorded. By default, WATCHER events are logged to the CENTRAL operator class.

QUALIFIERS

/FILE=file-spec

Directs event logging to the specified file.

/OPERATOR=oper-list

Specifies a list of one or more operator classes to which WATCHER events should be logged. If more than one operator class name is specified, the list should be comma-separated and surrounded by parentheses.

SET INSWAP

Sets the $GETJPI NOINSWAP control flag used by WATCHER.

FORMAT

SET [NO]INSWAP

DESCRIPTION

Under VAX/VMS V5.2 and later (and all versions of OpenVMS AXP), the $GETJPI system service supports a control flag that prevents it from taking any action that would result in the swapping in of a process that is currently swapped out. SET NOINSWAP enables the use of this control flag in WATCHER, causing WATCHER to ignore any swapped-out processes. The default setting is INSWAP.

Using SET NOINSWAP may result in inactive processes not being logged out by WATCHER, or in some cases may result in processes getting logged out prematurely, because activity of swapped-out processes cannot be determined when NOINSWAP is set.

SET INTERVAL

Sets the hibernation interval between processing passes.

FORMAT SET INTERVAL=delta-time

DESCRIPTION

This command sets the length of time WATCHER hibernates between processing passes. The default is 5 minutes. The value you should use should be smaller than the warning and logout intervals for all terminals and smaller than the difference between the logout and warning intervals for any single terminal. Too small a value, however, will cause WATCHER to waste CPU time.

SET MULTIWARN

Enables or disables multiple warnings.

FORMAT

SET [NO]MULTIWARN

Command Qualifiers	Defaults
/INTERVAL=delta-time	*/INTERVAL="0 00:05:00"*

DESCRIPTION

By default, WATCHER displays only one warning on terminals, at the time specified on the /WARNING qualifier. SET MULTIWARN enables multiple warnings; one at the /WARNING time and again every five minutes (or whatever interval you specify) until the /LOGOUT or /DISCONNECT time is reached.

This is a system-wide setting.

QUALIFIERS

/INTERVAL=delta-time

Specifies the interval of time that should occur between warnings. If omitted, defaults to five minutes. This value should equal or exceed the wakeup interval value (specified by SET INTERVAL). The actual interval between warnings may be longer than the specified time, since checks are made only at each processing pass (the interval between which is controlled by the wakeup interval value).

SET VERIFY

Enables or disables echoing of commands in command files.

FORMAT SET [NO]VERIFY

DESCRIPTION

The SET VERIFY command turns on command verification, so that commands read from WCP command files are echoed to the terminal. SET NOVERIFY turns off verification, which is the default.

SET WATCH_DEFAULT

Establishes defaults for subsequent WATCH commands.

FORMAT

SET WATCH_DEFAULT

Command Qualifiers	Defaults
/[NO]DISCONNECT[=deltatime] */[NO]FORCE_EXIT[=deltatime]* */[NO]LOGOUT[=deltatime]* */MEASURE=(measurement[,...])* */[NO]WARNING[=deltatime]*	

DESCRIPTION

The SET WATCH_DEFAULT command is used to set logout, warning, and measurement defaults for subsequent WATCH commands.

The defaults set by this command are saved between WCP sessions.

QUALIFIERS

/[NO]DISCONNECT[=deltatime]

Sets the default for disconnects, to be used if not specified on subsequent WATCH commands. /DISCONNECT, /FORCE_EXIT, and /LOGOUT are mutually exclusive.

/[NO]FORCE_EXIT[=deltatime]

Sets the default for forced image exits to be used on subsequent WATCH commands. /DISCONNECT, /FORCE_EXIT, and /LOGOUT are mutually exclusive.

/[NO]LOGOUT[=deltatime]

Sets the default for logouts, to be used if not specified on subsequent WATCH commands. /DISCONNECT, /FORCE_EXIT, and /LOGOUT are mutually exclusive.

/MEASURE=(measurement[,...])

Sets the default measurements to be used for activity determination if not specified on subsequent WATCH commands. For *measurement*, specify one of the following:

CPU[:*threshold*]	CPU time (the sum of the CPU time used by the process and all its subprocesses, in centiseconds) should be used as a criterion. If *threshold* is specified, the difference in CPU time between passes must exceed the specified threshold for a process to be considered active.

PROCESS_IO[:*threshold*]	Process I/O (the sum of the buffered and direct I/O counts for the process and all its subprocesses) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.
TERMINAL_IO[:*threshold*]	Terminal I/O (the operation count on the terminal device) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.

Any combination of PROCESS_IO, CPU, and TERMINAL_IO is permitted. If omitted, *threshold* values default to zero. Note that threshold values should be chosen as a function of the wakeup interval (defined with SET INTERVAL).

/[NO]WARNING[=deltatime]

Sets the defaults for warnings, to be used if not specified on subsequent WATCH commands.

SHOW

Displays all or part of the current configuration.

FORMAT

SHOW { *ACTION* | *ALL* | *BELL* | *DAYS* | *DEBUG* | *DEFAULTS* | *EVENT_LOG* | *EXCLUDE* | *FILE* | *GLOBALS* | *INSWAP* | *INTERVAL* | *MULTIWARN* | *OVERRIDE* | *WATCH* }

Command Qualifiers	Defaults
/[NO]COMMAND	*/NOCOMMAND*
/OUTPUT=file-spec	*/OUTPUT=SYS$OUTPUT:*

DESCRIPTION

The SHOW command displays information about the current configuration and the WCP default settings.

QUALIFIERS

/[NO]COMMAND

The /COMMAND qualifier indicates that the display should be formatted as the commands that would be entered to create the specified records. Use /COMMAND with the /OUTPUT qualifier to create an MCP command file that can be altered with your favorite editor, then read back into MCP to create a new configuration.

/OUTPUT=file-spec

The /OUTPUT qualifier is used to direct the SHOW result to a file or other device. By default, the result is displayed on the current output device, SYS$OUTPUT.

SHUTDOWN

Sends a shutdown command to the WATCHER process.

FORMAT SHUTDOWN

DESCRIPTION

This command sends a shutdown command to the WATCHER process, which causes WATCHER to close its log files and exit.

OPER and SYSPRV privileges are required for this command.

WATCH

Defines a watch rule.

FORMAT

WATCH *device-pat*

Command Qualifiers	Defaults
/ACCPORNAM=port-pat */DELETE* */[NO]DISCONNECT[=deltatime]* */[NO]FORCE_EXIT[=deltatime]* */[NO]LOGOUT[=deltatime]* */MEASURE=(measurement[,...])* */[NO]WARNING[=deltatime]*	*/ACCPORNAM=**

PARAMETERS

device-pat

A terminal device name or pattern containing wildcards, to identify the terminal(s) to be watched.

DESCRIPTION

This command is used to add or remove (with /DELETE) a watch rule to the WATCHER configuration. When WATCHER is running, a process running on any terminal matching the specified criteria will be watched for inactivity, using the specified parameters.

QUALIFIERS

/ACCPORNAM=port-pat

Port name or pattern containing wildcards, identifying the terminal port(s) (for terminal servers and other devices using port names) to be watched. The default is any port.

/DELETE

Specifies that the rule should be deleted from the configuration. All criteria must match exactly for the rule to be deleted.

/[NO]DISCONNECT[=deltatime]

For systems with virtual terminals enabled, this qualifier specifies whether the terminal should be disconnected from the system, and if so, how long the terminal should be inactive before the disconnection occurs. If virtual terminals are not enabled, or the terminal to be forced off is not connected through a virtual terminal, the process is logged out (the same effect as for the /LOGOUT qualifier). If both this qualifier and /LOGOUT are omitted, the disconnect/logout default is taken from the current SET WATCH_ DEFAULT setting.

The /DISCONNECT, /FORCE_EXIT, and /LOGOUT qualifiers are mutually exclusive.

Note: The terminal-disconnection code used by WATCHER runs in kernel mode, at elevated IPL, and hence may result in a system crash if the VMS terminal UCB extensions or terminal class driver changes. As with all code that uses undocumented VMS internals, exercise caution when using /DISCONNECT for the first time and when you upgrade to a new version of VMS.

The disconnect code is the only part of WATCHER that uses kernel mode.

/[NO]FORCE_EXIT[=deltatime]

Specifies that any user-mode image currently running at the terminal should be forced to exit, without actually logging the user off.

The /DISCONNECT, /FORCE_EXIT, and /LOGOUT qualifiers are mutually exclusive.

/[NO]LOGOUT[=deltatime]

Specifies whether the terminal should be logged out, and if so, how long the terminal should be inactive before logout occurs. If this qualifier and /DISCONNECT are omitted, the logout/disconnect default is taken from the current SET WATCH_DEFAULT setting.

The /DISCONNECT, /FORCE_EXIT, and /LOGOUT qualifiers are mutually exclusive.

/MEASURE=(measurement[,...])

Specifies the measurements that should be used for activity determination. For *measurement*, specify one of the following:

CPU[:*threshold*]	CPU time (the sum of the CPU time used by the process and all its subprocesses, in centiseconds) should be used as a criterion. If *threshold* is specified, the difference in CPU time between passes must exceed the specified threshold for a process to be considered active.
PROCESS_IO[:*threshold*]	Process I/O (the sum of the buffered and direct I/O counts for the process and all its subprocesses) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.
TERMINAL_IO[:*threshold*]	Terminal I/O (the operation count on the terminal device) should be used as a criterion. If *threshold* is specified, the difference in I/O counts between passes must exceed the specified threshold for a process to be considered active.

Any combination of PROCESS_IO, CPU, and TERMINAL_IO is permitted. If omitted, *threshold* values default to zero. Note that threshold values should be chosen as a function of the wakeup interval (defined with SET INTERVAL). If no /MEASUREMENT qualifier is specified, the measurement settings are taken from the SET WATCH_DEFAULT settings.

/[NO]WARNING[=deltatime]

Specifies whether the terminal should be warned about inactivity, and if so, how long the terminal should be inactive before warning is sent. The warning interval should be less than the logout interval. If omitted, the default is taken from the current SET WATCH_DEFAULT setting.