WATCH
Terminal Monitoring
Program for OpenVMS

Order Number: Documentation Order No. AD-120-3.4

This manual describes the installation and proper operation of the Watch Terminal Monitoring Program for OpenVMS.

Revision/Update Information:	This is a revised manual
Operating System and Version:	OpenVMS VAX Version 5.4 or later
Operating System and Version:	OpenVMS Alpha Version 6.1 or later
Software Version:	WATCH Version 3.4

Advanced Systems Concepts
Parsippany, New Jersey

The information in this document is subject to change without notice and should not be construed as a commitment by Advanced Systems Concepts, Inc. ASCI assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and may only be used or copied in accordance with the terms of such license.

DIGITAL, VAX, OpenVMS, are trademarks of Hewlett Packard.

ASCI, ASCI logo, and WATCH are trademarks of Advanced Systems Concepts, Inc.

Contents

1 Introduction - What is WATCH?

WATCH Version 3.4 is a utility which runs on OpenVMS VAX Version 5.2 (or later) and OpenVMS Alpha Version 6.1 (or later) and allows a user to monitor a terminal connected to any supported OpenVMS terminal class driver device (i.e., DZ, DMF, PSI, LAT). WATCH can enable a user (termed, the *watcher*) to watch the input and/or output streams of another terminal (termed, the *watchee*).

This is analogous to a car with two steering wheels; each terminal (the watcher and the watchee) may enter input into the watchee's input stream. WATCH's input capture may be disabled, in which case the watcher may only passively observe the watched terminal's output stream.

WATCH allows the output events to the terminal to be displayed on the terminal (the default) and/or logged to a disk file.

An interesting use of WATCH is to spawn it as a sub-process to passively watch your own terminal session. A complete output log of your terminal session can be transparently produced. In addition, WATCH will monitor the input stream for special "meta" characters which start and stop its disk logging activity. In this way, you can avoid capturing data you don't want to have captured.

2 WATCH Installation

2.1 Product Deliverables

Compare the contents of the WATCH distribution kit with the list which follows to determine if the kit you have received is complete.

WATCH Product Checklist

1. *Please Read Me First* Letter
2. **WATCH** User Manual
3. **WATCH** Distribution Media
4. **WATCH** Product Description
5. **ASCI** Software License Agreement

If the contents of the kit you have received are incomplete, please contact Advanced Systems Concepts, Inc.

2.2 Installing WATCH

The WATCH installation process uses the standard OpenVMS installation procedure - VMSINSTAL.

To begin the installation, physically mount the distribution media on any available device. Then, log into your system using the SYSTEM account and enter the following commands:

```
$ SET DEFAULT SYS$UPDATE
$ @VMSINSTAL
```

VMSINSTAL will verify that account you are using for the installation has the required privileges and quotas to ensure a successful installation.

VMSINSTAL will prompt for the location of the distribution kit and the name of the product to be installed. Consult the *Please Read Me first* Letter to determine the product name for the particular version of OpenVMS that you are running.

The WATCH product specific questions are:

```
* Does this product have an authorization key registered and loaded? YES
```

WATCH uses Digital's License Management Facility (LMF) to determine software license compliance and requires a valid Product Authorization Key (PAK) to be registered and loaded before WATCH can be used. The WATCH installation will complete successfully if you answer NO to this

question; however, you will not be able to start or use WATCH until a valid PAK has been registered and loaded.

To register the PAK, log into a privileged account (SYSTEM) and execute the VMSLICENSE.COM procedure located in the SYS$UPDATE directory. This procedure will prompt you for the various items found on your WATCH PAK.

For complete information on the use of the VMSLICENSE procedure and LMF, see the *OpenVMS License Management Facility Manual*.

```
    WATCH V3.4 requires a location for its files.
    Please enter a device & directory for WATCH V3.4
* Default is [SYS$SYSDEVICE:[WATCH]]:
```

The WATCH installation will place the product components in the directory SYS$SYSDEVICE:[WATCH]. You may install WATCH in an alternate directory or on an alternate device by supplying a specification for the alternate location when prompted with the above installation message.

```
* Do you want WATCH help placed in system-wide help library?
```

If you wish, you may elect to have the WATCH help text added to your normal OpenVMS help library. The default answer to this question is YES.

```
* Do you want to add WATCH to your system command tables?
```

If you wish, you may elect to have the the WATCH command added to to your system's system-wide command tables (DCLTABLES.EXE). The default answer to this question is YES. If you enter NO then, a SET COMMAND must be issued prior to using WATCH.

```
* Will you permit WATCH Notification to be disable with /NONOTIFY?
```

By default, WATCH will notify the process to be WATCHed that it is being WATCHed. If you wish, you may elect to allow notification to be disabled through the use of the WATCH command line qualifier, /NONOTIFY. The default answer to this question is NO. If you enter YES then, WATCH notification may be disabled by processes which hold the appropriate WATCH rights.

Note: In certain countries, there is a legal requirement to *always* notify a user when they are being watched. By answering NO to this question, you will ensure that the user will always be notified.

```
* Will you permit the use of WATCH for SELF watching as a default?
```

By default, WATCH can not be used for self watching or watching any processes with the same UIC unless the process hold the appropriate rightlist identifier. If you wish, you may elect allow WATCH to be used for SELF watching and watching processes with the same UIC as that of the watcher.

By default, the answer to this question is NO. If you enter YES then, the WATCH$OWNER identifer is assumed to be held by all processes on the system.

2.3 Installing WATCH (Post-Installation)

After the WATCH product has been successfully installed and the WATCH components have been moved to their target directories, VMSINSTAL invokes the post-installation phase of the WATCH installation. The WATCH rightslist identifiers are defined as part of the post-intallation phase of the product's installation. The WATCH identifiers will be added to or updated in your system's rightslist database. After the definition of the rightslist identifiers, the System Manager is then responsible for granting the rights to use the various features of WATCH to users on the system.

For more information on the control of WATCH features, WATCH rights, and granting and revoking WATCH identifiers, refer to the chapter: **How to Use WATCH**.

2.4 Sample WATCH Installation

The following section contains a sample **WATCH** installation depicting the various VMSINSTAL prompts and product queries which would be encountered in a typical installation.

Figure 2–1 Sample WATCH Installation

```
$ SET UIC [1,4]
$ SET DEFAULT SYS$UPDATE
$ @SYS$UPDATE:VMSINSTAL

 VAX/VMS Software Product Installation Procedure V6.0

It is 9-AUG-1995 at 14:08.

Enter a question mark (?) at any time for help.
* Are you satisfied with the backup of your system disk [YES]? YES
* Where will the distribution volumes be mounted: $1$MKA500:

Enter the products to be processed from the first distribution volume set.
* Products: WATCH034
* Enter installation options you wish to use (none):
The following products will be processed:

  WATCH V3.4

 Beginning installation of WATCH V3.4 at 14:09

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

  WATCH V3.4 Installation Procedure
  Copyright (C) 1984,1988,1990,1994,1995 by Advanced Systems Concepts, Inc.

  ******************************************************************************
  Attention -  System Manager / Installer:

  1. Please remember to add SYS$STARTUP:WATCH_STARTUP.COM to the OpenVMS
     Startup procedures.  SYS$STARTUP:WATCH_STARTUP.COM must be executed
     before WATCH can be used.

  2. WATCH Release Notes can be found in SYS$HELP.

  3. You may elect to add the WATCH command to your DCL Tables.

  4. You may elect to add WATCH Help to your VMS Help library.

  5. This product requires that a WATCH license be registered and loaded
     on the system prior to using the product.
  ******************************************************************************

* Do you want to purge files replaced by this installation [YES]? yes

        Product:      WATCH
        Producer:     ASCI
        Version:      3.4
        Release Date: 1-SEP-1991

* Does this product have an authorization key registered and loaded? YES

    WATCH V3.4 requires a location for its files.
    Please enter a device & directory for WATCH V3.4

* Default is [SYS$SYSDEVICE:[WATCH]]:
%VMSINSTAL-I-SYSDIR, This product creates system disk directory  SYS$SYSDEVICE:[WAT

* Do you want WATCH help placed in system-wide help library [YES]? ?
```

Figure 2–1 Cont'd on next page

Figure 2–1 (Cont.) Sample WATCH Installation

```
  You may elect to have the WATCH help text added to the system-wide help
  library (SYS$HELP:HELPLIB.HLB) making the WATCH help text available to
  all users by default.

  If you do not add WATCH help to the system-wide help library, users may
  access WATCH help by creating a WATCH help library and then, define the
  logical HLP$LIBRARY to the location of this help library file.

  For additional information, refer to the OpenVMS Library Utility Manual.

* Do you want WATCH help placed in system-wide help library [YES]? YES

* Do you want to add WATCH to your system command tables [YES]? ?

  You may elect to have the WATCH command added to the system-wide command
  tables (SYS$LIBRARY:DCLTABLES.EXE) making the WATCH command available to
  all users by default.

  If you do not add WATCH to the system-wide command tables, any user that
  needs to run WATCH must add the command to their process command tables
  using the command:

          $ SET COMMAND WATCH_PRODUCT:WATCH.CLD

  Prior to invoking WATCH.

* Do you want to add WATCH to your system command tables [YES]? YES


  You may elect to allow the notification by WATCH to be disabled with the
  /NONOTIFY qualifier for processes which hold the 'WATCH$ANNOUNCE' rights
  identifier.

  As its default, WATCH notification can not be disabled even if a process
  holds the WATCH$ANNOUNCE identifier.  If you answer YES to this question
  you will enable the WATCH$ANNOUNCE identifier and may then grant this to
  a process, permitting it to disable WATCH notification.

  ** NOTE:  In certain countries, there is a legal requirement to _ALWAYS_
            notify a user when they are being watched.  If this is a legal
            requirement in your country, answer NO to this question.

* Will you permit WATCH Notification to be disabled with /NONOTIFY [NO]? NO


  You may elect to allow WATCH to be used (by default) by all users on your
  system to WATCH their own processes without requiring the process to hold
  the WATCH$OWNER identifier.

  As its default, WATCH will require a process to hold WATCH$OWNER to watch
  itself or any processes with the same UIC.

  If you answer YES to this question, WATCH$OWNER rights identifier will be
  assumed to be present by default.  This will disable your ability to dis-
  able (restrict) watch use (on a per-process basis) for self watching.

* Will you permit the use of WATCH for SELF watching as a default [NO]? NO
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

  WATCH V3.4 Post-Installation Procedure
  Copyright (C) 1984,1988,1990,1994,1995 by Advanced Systems Concepts, Inc.

  The following WATCH Rightslist Identifiers will now be defined in the
  system's Rightslist database.

  WATCH$OWNER ......... Right to WATCH processes with same owner UIC
  WATCH$GROUP ......... Right to WATCH processes with same group UIC
  WATCH$WORLD ......... Right to WATCH processes with any system UIC
```

Figure 2–1 Cont'd on next page

Figure 2–1 (Cont.) Sample WATCH Installation

```
  WATCH$CONTROL ....... Right to CONTROL/INPUT/ADVISE WATCHed terminal
  WATCH$NOLOGIN ....... Right to WATCH logged out (no owner) terminals
  WATCH$ANNOUNCE ...... Right to disable WATCHed terminal notification

  WATCH$INFORM ........ Right to be notified when process is WATCHed
  WATCH$BLOCK ......... Right to block attempts to WATCH the process
  WATCH$BLOCK_CONTROL . Right to block WATCH CONTROL/INPUT/ADVISE use

  WATCH$SUPERVISOR .... WATCH$CONTROL and WATCH$ANNOUNCE rights plus,
                        the right to override WATCH$INFORM, WATCH$BLOCK
                        and WATCH$BLOCK_CONTROL rights.

* Press return to contiune and add the identifiers to the system:
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$OWNER.
%UAF-I-RDBADDMSG, identifier WATCH$OWNER value: %X8001003E added to rights data bas
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$GROUP.
%UAF-I-RDBADDMSG, identifier WATCH$GROUP value: %X8001003F added to rights data bas
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$WORLD.
%UAF-I-RDBADDMSG, identifier WATCH$WORLD value: %X80010040 added to rights data bas
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$CONTROL.
%UAF-I-RDBADDMSG, identifier WATCH$CONTROL value: %X80010041 added to rights data b
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$ANNOUNCE.
%UAF-I-RDBADDMSG, identifier WATCH$ANNOUNCE value: %X80010042 added to rights data
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$SUPERVISOR.
%UAF-I-RDBADDMSG, identifier WATCH$SUPERVISOR value: %X80010043 added to rights dat
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$BLOCK.
%UAF-I-RDBADDMSG, identifier WATCH$BLOCK value: %X80010044 added to rights data bas
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$BLOCK_CONTRO
%UAF-I-RDBADDMSG, identifier WATCH$BLOCK_CONTROL value: %X80010045 added to rights
%VMSINSTAL-I-ACCOUNT, This installation adds an identifier named WATCH$INFORM.
%UAF-I-RDBADDMSG, identifier WATCH$INFORM value: %X80010046 added to rights data ba

 Installation of WATCH V3.4 completed at 14:15

Enter the products to be processed from the next distribution volume set.
* Products: EXIT

 VMSINSTAL procedure done at 14:16

$
```

After WATCH has been installed, please include the command procedure:

```
@SYS$STARTUP:WATCH_STARTUP
```

as part of your system startup.

3 How to Use WATCH

3.1 Invoking WATCH

If, during the product installation, the **WATCH** command was not added to the system-wide command table then, prior to invoking WATCH, you must enter the following command:

```
$ SET COMMAND SYS$SYSTEM:WATCH
```

This will add the WATCH command syntax to the command tables of your current process.

WATCH is invoked by the DCL command WATCH.

When the WATCH command is issued, the program will continue to execute your request until you have terminated its function. In an interactive process, this is done by typing a special escape key (default is CTRL/A) on the watching terminal. This will cause WATCH to cease watching the specified terminal and return the terminal to its own process and back to DCL command level, release the watched terminal from watch mode.

It is also possible to run WATCH as a detached or sub-process. If this method of watching a terminal is desired, WATCH must be invoked and terminated by special procedures. These procedures are discussed in the section titled, "Invoking and Terminating a Non-interactive WATCH Process."

WATCH requires CMKRNL, WORLD, OPER and SYSPRV privileges to executed. WATCH is installed with these privileges as part of the WATCH_STARTUP.COM procedure.

In addition to installing the WATCHMAN.EXE image with the required privileges, the startup procedure also loads the WATCH support execlet. When the startup procedure is invoked for the first time after a system startup, a message similar to the following will be displayed:

```
$ @SYS$STARTUP:WATCH_STARTUP

ASCI WATCH_EXECLET loaded at 802E7800
                             8047D300
                             00000003
```

If the support execlet does not load, WATCH can not be used. If the WATCH execlet fails to load, check to see that the WATCH license has been registered and loaded.

3.2 Product Use Security

WATCH security is controlled by a group of rightslist identifiers which are normally defined during the product installation. At the System Manager's discretion, a process may be granted one or more of these WATCH identifiers to enable the process to WATCH another terminal and perform various WATCH functions.

WATCH rightslist identifiers, when granted to a process, will enable the process holding the identifier to utilize the features of WATCH which are governed by that identifier. A process which does not hold a particular WATCH identifier will not be able to access or utilize the WATCH features which the identifier represents. The System Manager controlling these rights should fully understand the repercussions of each WATCH identifier before granting a process any identifiers.

Because rights are granted to UICs and not to user accounts, care should be taken to insure that a UIC which is granted an identifier is unique to a username. In addition, the System Manager granting WATCH rights should also be aware of the repercussions of granting a WATCH identifier in conjunction with the use of the SET UIC command.

3.3 WATCH Rightslist Identifiers

Ten (10) rightslist identifiers have been identified for use with **WATCH**. Table 3–1 describes these ten identifiers. Section 3.3.1 contains details on each of the listed identifiers and discusses the security implications of each identifier.

Table 3–1 WATCH Rightslist Identifiers

WATCH Identifier	Brief Description
WATCH$OWNER	Right to WATCH processes with same owner UIC
WATCH$GROUP	Right to WATCH processes with same group UIC
WATCH$WORLD	Right to WATCH processes with any system UIC
WATCH$CONTROL	Right to CONTROL/INPUT/ADVISE WATCHed terminal
WATCH$NOLOGIN	Right to WATCH logged out (no owner) terminals
WATCH$ANNOUNCE	Right to disable WATCHed terminal notification
WATCH$INFORM	Right to be notified when process is WATCHed
WATCH$BLOCK	Right to block attempts to WATCH the process
WATCH$BLOCK_CONTROL	Right to block WATCH CONTROL/INPUT/ADVISE use

Table 3–1 (Cont.) WATCH Rightslist Identifiers

WATCH Identifier	Brief Description
WATCH$SUPERVISOR	WATCH$CONTROL and WATCH$ANNOUNCE rights plus the right to override WATCH$INFORM, WATCH$BLOCK and WATCH$BLOCK_CONTROL rights.

3.3.1 WATCH Identifier Specifics

The following sections will describe the WATCH rightslist identifiers listed in Table 3–1 in detail and will discuss the security implications of each identifer.

3.3.1.1 The WATCH$OWNER Identifier

The WATCH$OWNER identifier grants its holder the right to use WATCH to access terminals which are owned by processes with the same UIC as that of the watcher.

During product installation, the installer may elect to have this identifier be considered *granted by default* for all processes. If this is the case, the WATCH$OWNER identifier is assumed to be held by every process on the system. If the installer did not elect to make WATCH$OWNER a default, this identifier must be granted on an individual basis for a process to have the right to watch its own terminal(s).

WATCH$OWNER is one of three identifiers, the other two being WATCH$GROUP and WATCH$WORLD, used to determine the scope of access to process owned terminals. By itself, the WATCH$OWNER identifier grants its holder the ability to watch a terminal owned by a process with the same UIC; however, it can not control or advise the terminal.

3.3.1.2 The WATCH$GROUP Identifier

The WATCH$GROUP identifier grants its holder the right to use WATCH to access terminals which are owned by processes with the same group UIC as that of the watcher. This identifier must be granted to a process in order for that process to have the right to watch terminals owned by processes in its own group.

WATCH$GROUP is one of three identifiers, the other two being WATCH$OWNER and WATCH$WORLD, used to determine the scope of access to process owned terminals. By itself, the WATCH$GROUP identifier grants its holder the ability to watch terminals owned by a process with the same group UIC; however, it can not control or advise those terminals.

3.3.1.3 The WATCH$WORLD Identifier

The WATCH$WORLD identifier grants its holder the right to use WATCH to access terminals which are owned by processes with any group and owner UIC. This identifier must be granted to a process in order for that process to have the right to watch terminals owned by any process.

WATCH$WORLD is one of three identifiers, the other two being WATCH$OWNER and WATCH$GROUP, used to determine the scope of access to process owned terminals. By itself, the WATCH$WORLD identifier grants its holder the ability to watch terminals owned by any process on the system; however, it can not control or advise those terminals.

3.3.1.4 The WATCH$CONTROL Identifier

The WATCH$CONTROL identifier grants its holder the right to use WATCH to control or advise terminals which it also has the right to watch. This identifier must be granted to a process in order that the process have the right to input data to the terminal.

3.3.1.5 The WATCH$ANNOUNCE Identifier

The WATCH$ANNOUNCE identifier grants its holder the right to use WATCH without notifying the terminal being watched that it is being watched.

A process granted this identifier can specify the WATCH command qualifier /NONOTIFY to disable the notification message which is broadcast to the watched terminal. Without this identifier, a process specifying the /NONOTIFY qualifier will have the qualifier ignored and the notification message will be broadcast.

Note: In certain countries, there is a legal requirement to *always* notify a user when they are being watched. During product installation, the installer may have elected to *always* have WATCH notify the target that it is being watched. If this is the case, the WATCH$ANNOUNCE identifier and the /NONOTIFY qualifier have no function.

3.3.1.6 The WATCH$NOLOGIN Identifier

The WATCH$NOLOGIN identifier grants its holder the right to use WATCH a terminal which is not currently owned by a process; that is, a terminal which no user is logged in to when the WATCH command is issued.

This identifier can be used to WATCH a terminal prior to a user logging in and should not be granted without consideration to this privilege.

3.3.1.7 The WATCH$INFORM Identifier

The WATCH$INFORM identifier grants its holder the right to be notified when WATCH is in use to watch its terminal. This identifer applies to the process being watched and overrides the watchers ability to disable WATCH notification when holding the WATCH$ANNOUNCE identifier.

Only processes holding WATCH$SUPERVISOR right can circumvent the notification asserted by the WATCH$INFORM identifier.

Note: In certain countries, there is a legal requirement to *always* notify a user when they are being watched. During product installation, the installer may have elected to *always* have WATCH notify the target that it is being watched. If this is the case, the WATCH$SUPERVISOR and the /NONOTIFY qualifier have no function.

3.3.1.8 The WATCH$BLOCK Identifier

The WATCH$BLOCK identifier grants its holder the right to completely block the use of WATCH to watch its terminal. This identifer applies to the process being watched and overrides the watchers ability to WATCH the terminal, regardless of its access rights.

Only processes holding WATCH$SUPERVISOR right can circumvent the blocking asserted by the WATCH$BLOCK identifier.

3.3.1.9 The WATCH$BLOCK_CONTROL Identifier

The WATCH$BLOCK_CONTROL identifier grants its holder the right to block the use of WATCH to input data to its terminal; the watcher is still permitted to watch the terminal. This identifer applies to the process being watched and overrides the watchers ability to input, control or advise the terminal, regardless of its access rights.

Only processes holding WATCH$SUPERVISOR right can circumvent the blocking asserted by the WATCH$BLOCK_IDENTIFIER identifier.

3.3.1.10 The WATCH$SUPERVISOR Identifier

The WATCH$SUPERVISOR identifier grants its holder the rights associated with the identifiers: WATCH$CONTROL and WATCH$ANNOUNCE. In addition, a process holding this identifier can override the WATCH rights of the target terminal's process; namely, WATCH$BLOCK_CONTROL, WATCH$BLOCK and WATCH$INFORM.

A process holding the WATCH$SUPERVISOR identifier can only apply the rights granted by this identifier to terminals owned by processes which it has the rights to watch. These are granted by the identifiers: WATCH$OWNER, WATCH$GROUP and WATCH$WORLD.

Note: In certain countries, there is a legal requirement to *always* notify a user when they are being watched. During product installation, the installer may have elected to *always* have WATCH notify the target that it is being watched. If this is the case, the WATCH$SUPERVISOR and the /NONOTIFY qualifier have no function.

3.3.2 Granting and Revoking WATCH Identifiers

The WATCH rightslist identifiers are granted to and revoked from a user through the use of the AUTHORIZE utility's GRANT/IDENTIFIER and REVOKE/IDENTIFIER commands. Identifiers rights held by a process are immediately recognized by WATCH when they are GRANTED to or REVOKED from a process. This allow the system manager to temporarily GRANT WATCH identifier rights to a process for short durations and then, REVOKE them without relying on the user having to log out and log in to enable or disable the rights.

The WATCH identifiers can be displayed using the following AUTHORIZE utility commands:

```
UAF> SHOW/IDENTIFIER WATCH$GROUP
  Name                            Value           Attributes
  WATCH$GROUP                     %X80010034

UAF> SHOW/IDENTIFIER *
    ...
    ...
  WATCH$ANNOUNCE                  %X80010037      NOACCESS
  WATCH$BLOCK                     %X80010039
  WATCH$BLOCK_CONTROL             %X8001003A
  WATCH$CONTROL                   %X80010036
  WATCH$GROUP                     %X80010034
  WATCH$INFORM                    %X8001003B
  WATCH$OWNER                     %X80010033
  WATCH$SUPERVISOR                %X80010038
  WATCH$WORLD                     %X80010035
    ...
    ...
```

The WATCH rightslist identifiers are normally defined as part of the product installation during the post-installation phase. If the WATCH rightslist identifiers are not defined in the system's rightslist database, they can be defined using the following AUTHORIZE utility command:

```
UAF> ADD/IDENTIFIER WATCH$OWNER
%UAF-I-RDBADDMSG, identifier WATCH$OWNER value: %X80010033 added to rights database
```

If desired, the identifiers can also be defined by re-installing the WATCH product from the distribution kit. This is the recommended method for defining the WATCH identifiers because it will also set certain WATCH identifier attributes based on installation options.

A user is granted a WATCH identifier right using the following AUTHORIZE utility command:

```
UAF> GRANT/IDENTIFIER WATCH$OWNER A_USER
%UAF-I-GRANTMSG, identifier WATCH$OWNER granted to A_USER
```

A WATCH identifier right can be revoked from a user using the following AUTHORIZE utility command:

```
UAF> REVOKE/IDENTIFIER WATCH$OWNER A_USER
%UAF-I-REVOKEMSG, identifier WATCH$OWNER revoked from A_USER
```

Several WATCH identifiers can be granted to a process to enable the holder to perform various WATCH functions and/or enable WATCH features.

For example, issuing the following AUTHORIZE utility GRANT commands will enable the user A_MANAGER to watch and advise all the members of the group which A_MANAGER is a member. In addition, A_MANAGER is granted the ability to disable WATCH notification using the /NONOTIFY qualifier because of the WATCH$SUPERVISOR identifier. Holding the WATCH$SUPERVISOR right, A_MANAGER can also circumvent WATCH rights such as: WATCH$BLOCK and WATCH$INFORM, that may be held by members of the same group.

```
UAF> GRANT/IDENTIFIER WATCH$GROUP A_MANAGER
%UAF-I-GRANTMSG, identifier WATCH$GROUP granted to A_MANAGER
UAF> GRANT/IDENTIFIER WATCH$SUPERVISOR A_MANAGER
%UAF-I-GRANTMSG, identifier WATCH$OWNER granted to A_MANAGER
```

Because the rights granted by the WATCH$SUPERVISOR identifier are limited by the scope of access granted by WATCH$GROUP, A_MANAGER can not use WATCH to *spy* on users outside of the group.

3.4 WATCH Command Summary

The subsequent section presents a summary of the WATCH command syntax. WATCH adheres to the syntax and grammar rules of DCL commands as described in the OpenVMS Command Language User's Guide.

WATCH Command

This command is used to intercept a terminal's input and/or output streams.

FORMAT

WATCH Terminal-device

Command Qualifiers	Defaults
/APPEND	*See text*
/ESCAPE=value	*CTRL/A*
/[NO]INPUT	*/NOINPUT*
/[NO]MESSAGE	*/MESSAGE*
/[NO]NOTIFY	*/NOTIFY*
/[NO]OUTPUT[=file-spec]	*/NOOUTPUT*
/[NO]START	*/START*
/WATCH	*See text*

restrictions

You may never watch your own terminal in an interactive session. You may do so as a detached or sub-process, see the section titled "Invoking and Terminating a Non-interactive WATCH process."

prompts

Terminal Device: terminal-device

PARAMETERS

terminal-device

Specifies the terminal device that you wish to watch.

COMMAND QUALIFIERS

/APPEND

This qualifier is used with /OUTPUT to indicate that you wish to append this log of recorded terminal events with the log of a previous WATCH session.

/ESCAPE

This qualifier is used when WATCH is invoked interactively and specifies a decimal value to represent an "escape from WATCH" sequence.

By default, the escape character is CTRL/A (decimal value 1). When CTRL/A (or your specified key) is pressed, WATCH will release the watched terminal and return to DCL.

/INPUT
/NOINPUT

This qualifier is used to enable or disable terminal input interception. **/CONTROL** and **/ADVISE** are synonyms for the **/INPUT** qualifier.

By default, /NOINPUT is enabled when WATCH is executed. WATCH$CONTROL or WATCH$SUPERVISOR rights are required to enable input with /INPUT.

/MESSAGE
/NOMESSAGE

This qualifier allows WATCH to display its informational messages on your terminal. These messages usually indicate what is being watched, where data is being logged, and how to exit WATCH.

By default, /MESSAGE enables the display of informational messages.

/NOMESSAGE prevents the informational messages from being displayed. This may be useful when watching your own terminal and logging the data to disk. In that case, you may not want the messages entered into the log.

Fatal error messages are always displayed.

/NOTIFY
/NONOTIFY

This qualifier is used to enable or disable notification to the terminal being watched that WATCH is being used to observe the terminal.

By default, /NOTIFY is enabled. WATCH$ANNOUNCE or WATCH$SUPERVISOR rights are required to use the /NONOTIFY qualifier to disable notification.

/OUTPUT[=file-spec]
/NOOUTPUT

Indicates whether intercepted output is to be written to a disk file.

By default, ouput is not written to a disk file. If /OUTPUT is specified, an optional file-specification may be added which indicates where the output is to be written. If /OUTPUT is specified, and no file specification indicated, the default file specification of WATCH.OUT is used.

/START
/NOSTART

This qualifier is used with /OUTPUT and tells WATCH whether or not to begin logging to disk immediately. /START is the default. If specified, WATCH will begin disk logging as soon as it is activated. /NOSTART will cause WATCH to wait until it is told to begin recording. It is told to do so by entering ~L (Tilde L) on the watching terminal. It will stop disk logging when ~N (Tilde N) is entered. In this way, selected portions of a terminal session may be recorded. Since WATCH reads each character in real time, the "~L" and "~N" commands will be acted upon immediately. It is not necessary to enter a delimiter. You may even use the DELETE key () to rub them out.

/WATCH

This qualifier is used with /OUTPUT to indicate that you wish to watch the events that are being logged to disk. By default, terminal events are not logged to the watcher's terminal when /OUTPUT is used.

DESCRIPTION

The **WATCH** command allows a user at one terminal to watch another terminal session. The terminal device specified must use the OpenVMS class driver mechanism. DECnet Remote terminals (RT devices) are not supported. PSI, LAT and DECterm terminals, as well as, various single and multi-line communications devices are supported. Most third-party TCP/IP NTY devices also use the OpenVMS class driver mechanism, permitting those terminal to be used with **WATCH** as well.

This command requires **CMKRNL** and **WORLD** privileges for proper operation. These are the normal installed privileges for **WATCH**.

EXAMPLES

1
```
$ WATCH TTA3
```

The WATCH command will intercept all terminal activity on TTA3 and write all events to the watcher's terminal. Terminal TTA3 will be

2
```
$ WATCH/OUTPUT=TTA3.LOG TTA3
```

This WATCH command will log all terminal events for TTA3 into a disk file named "TTA3.LOG" in your account. Nothing will be displayed on your terminal. Terminal TTA3 will be notified that it is being watched.

3
```
$ WATCH TTA3/OUTPUT=TTA3.LOG/WATCH
```

This command performs the functions of the previous two commands.

4
```
$ WATCH TTA3/INPUT
```

The WATCH command will intercept all terminal activity on TTA3 and write all events to the watcher's terminal. In addition, terminal control can also be used to input data as if it was actually being entered at the watched terminal's keyboard—assuming that the user holds the WATCH$CONTROL identifier.

3.5 Invoking and Terminating a Non-interactive WATCH Process

Normally, WATCH would be used interactively, using the WATCH command, to view and/or control the activity of another terminal. However, it is also possible to run WATCH in a sub-process. For example, to watch your own terminal in order to log the session activity to a file.

The WATCH.COM command procedure, found in WATCH_PRODUCT, allows you to spawn WATCH in a subprocess. To do so, enter the following command:

```
$ SPAWN/NOWAIT @WATCH_PRODUCT:WATCH terminal /qualifiers
```

(Note that a blank space must be entered after "terminal". This is to pass the qualifiers to the command procedure as a parameter.)

When WATCH invoked in this manner, it can not be terminated via a CTRL/A sequence. In order to terminate the spawned WATCH session, the STOP_WATCH command must be used. To use STOP_WATCH, enter the following commands:

- $ STOP_WATCH :== "$WATCH_PRODUCT:STOP_WATCH"
- $ STOP_WATCH <process_name>

<process_name> is the name given to the spawned WATCH process when it was created. If you SPAWNed the process, it should have a name of "Username_process-number".

3.6 On-Line HELP

A complete on-line help facility is provided with WATCH. It can be included in your system help when WATCH is installed if your System Manager elected this option. To invoke it, simply enter:

```
$ HELP WATCH
```

on your terminal. This will display a help text similar to that for other DCL commands. With it, you may obtain help on any or all WATCH command components.

4 WATCH Messages

The following list comprises all of the messages generated by WATCH (informational as well as error messages). Several WATCH messages may be accompanied by a VMS system error message which further describes the problem encountered.

BLOCKALL, Target holds WATCH$????? rights. Watch is blocked.

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process which owns the terminal to be watched holds rights preventing the use of WATCH to view the terminal. The WATCH identifier held by the process is listed in the message.

This message will be followed by a NOPRIV error message and the WATCH session will be terminated.

User Action: Verify the terminal device to be watched matched the rights you hold for invoking WATCH.

BLOCKINP, Target holds WATCH$????? rights. Input is blocked.

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process which owns the terminal to be watched holds rights preventing the use of WATCH to input data to the terminal. The WATCH identifier held by the process is listed in the message.

WATCH will continue with input capability disabled.

User Action: None.

DCLEXH, $DCLEXH error

Facility: WATCH

Severity: Fatal

Explanation: An error occurred when WATCH attemtped to establish its exit handler.

User Action: ASCI warranty or remedial support required.

DOINFORM, Target holds WATCH$????? rights. Notification enabled.

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process which owns the terminal to be watched holds rights overriding the /NONOITFY function. The WATCH identifier held by the process is listed in the message.

WATCH will continue and will notify the terminal to be watched.

User Action: None.

GETDVI, $GETDVI error

Facility: WATCH

Severity: Fatal

Explanation: An error occurred while attempting to obtain the specified terminal's attributes. An OpenVMS error message will accompany this error.

User Action: Probable cause: an unknown device was specified. Verify that the device entered exists.

GETJPI, $GETJPI error

Facility: WATCH

Severity: Fatal

Explanation: An error occurred while attempting to obtain process attributes.

User Action: Check privileges and quotas and determine if the WATCHMAN.EXE image was install and, if so, properly with the required privileges: CMKRNL, WORLD, OPER and SYSPRV.

GETVM, LIB$GET_VM Error

Facility: WATCH

Severity: Fatal

Explanation: An error occurred during WATCH activation, when WATCH is attempting to expand its process virtual memory. An OpenVMS error message accompany this error.

User Action: Ensure PGFILQUOTA and VIRTUALPAGCNT quota and SYSGEN parameter are sufficient for normal processing.

INIT_ERR, Initialization Error

Facility: WATCH

Severity: Fatal

Explanation: This error is printed along with one or more messages to indicate the error occurred while in initialization.

User Action: Verify the terminal device to be watched and your privileges for invoking WATCH.

INSFRIGHT, WATCH$????? rights required to watch terminal

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process holds insufficient WATCH rights to watch the terminal specified. The WATCH identifier that would, minimally, be required is listed in the message.

This message will be followed by a NOPRIV error message and the WATCH session will be terminated.

User Action: Verify the terminal device to be watched matched the rights you hold for invoking WATCH.

LKWSET, $LKWSET Error

Facility: WATCH

Severity: Fatal

Explanation: An error occurred while attempting lock WATCH into the process working set. An OpenVMS error message will accompany this error.

User Action: Check your privileges and determine if the WATCHMAN.EXE image was install and, if so, properly with the required privileges: CMKRNL, WORLD, OPER and SYSPRV.

LOGERR, An error occurred opening/creating a terminal log file

Facility: WATCH

Severity: Fatal

Explanation: This error occurs when WATCH encounters an error while accessing the specified terminal log file. An RMS error message should accompany this error.

User Action: Correct the problem after reading the alternate message text.

NOINRIGHT, WATCH$????? rights required. Input is disabled.

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process holds insufficient WATCH rights to use the /INPUT qualifier. The WATCH identifier required is listed in the message.

WATCH will continue with inputting disabled.

User Action: None.

NOPRIV, No Privilege to WATCH other terminal.

Facility: WATCH

Severity: Fatal

Explanation: This message is returned to indicate that the process does not have the privileges required to WATCH the specified terminal. This message is usually preceeded by a warning message detailing the necessary privilege or rights needed.

The WATCH session will be terminated.

User Action: Verify the terminal device to be watched matched the rights you hold for invoking WATCH.

NOSILENT, WATCH$????? rights required. Notification enabled.

Facility: WATCH

Severity: Warning

Explanation: This warning is returned to indicate that the process holds insufficient WATCH rights to use the /NONOTIFY qualifier. The WATCH identifier required is listed in the message.

WATCH will continue and will notify the terminal to be watched.

User Action: None.

NOSUPPORT, Terminal type not supported by WATCH

Facility: WATCH

Severity: Fatal

Explanation: This error occurs if the terminal to be watched is not a terminal class device or if the watcher's terminal is being watched by someone else (e.g., WATCH, PSS). This error will occur if you try to watch an RT (DECnet) device.

User Action: Verify actions.

NOTERM, Device specified is not a terminal

Facility: WATCH

Severity: Fatal

Explanation: WATCH can only watch terminals.

User Action: Specify a WATCH supported terminal device.

NOXPAG, non-existent page

Facility: WATCH

Severity: Fatal

Explanation: This message is return during WATCH initial activation and indicates that the WATCH support execlet has not been loaded. WATCH can only watch terminals.

User Action: Verify that the WATCH$STARTUP.COM procedure has been executed.

PUTERR, An error occurred during a $PUT operation

Facility: WATCH

Severity: Warning

Explanation: This warning message is printed, along with an RMS error message, to indicate some unexpected condition occurred which prevented you from writing the output record. WATCH will continue to execute but the data has been lost.

User Action: Verify that sufficent free disk space still exists.

WATCHED, Terminal in WATCH mode or being watched

Facility: WATCH

Severity: Fatal

Explanation: This error relates to the terminal specified in the WATCH command. That terminal may be using WATCH or PSS and, therefore, cannot be watched.

User Action: Verify actions and resubmit.

The following messages will be displayed when WATCH is started, unless /NOMESSAGE is specified.

LOGGING, Events on terminal "device" are being logged to "file"

Facility: WATCH

Severity: Success

Explanation: This message is displayed when you are logging your WATCH session to disk. "Device" is the terminal device you are watching and "file" is the log file specification.

User Action: None.

TERMINAL, Terminal "device" is being WATCHed on your terminal

Facility: WATCH

Severity: Success

Explanation: This message is displayed when you are in simple WATCH mode and are not logging to disk. "Device" is the terminal device you are watching.

User Action: None.

TOEXITA, Type "Escape_Sequence" to stop WATCH mode

Facility: WATCH

Severity: Success

Explanation: This message is printed to remind you of the proper way to terminate WATCH based on your run-time attributes. By default, the "Escape_Sequence" is the pressing CTRL/A to release WATCH.

User Action: None.

TOEXITB, Use STOP_WATCH command to stop WATCH mode

Facility: WATCH

Severity: Success

Explanation: This message is printed to remind you that WATCH is running with either input disabled and/or as a sub-process. CRTL/A will not cause WATCH to exit but rather use of the STOP_WATCH command will.

User Action: None.

RELEASE, Terminal "Device" is being released from WATCH

Facility: WATCH

Severity: Success

Explanation: This message is printed when you have successfully terminated WATCH. "Device" is the terminal that you were watching.

User Action: None.