Assassin User's Guide

June 2000

This document describes how to use the Assassin system software product to automatically manage idle user processes on OpenVMS computer systems. It is intended for system managers of all levels of experience.

Revision/Update Information: This manual supersedes previous versions of the *Assassin User's Guide*

Software Version: Assassin Version 5.2

WDA System Software
Santa Clara, California

June 2000

The information in this document is subject to change without notice and should not be construed as a commitment by WDA System Software. WDA System Software 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 be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment not supplied by Digital Equipment Corporation, or in conjunction with software other than the designated version of the OpenVMS Operating System.

The READER'S COMMENT form in the Forms Section of this document requests the user's critical evaluation of Assassin and the *Assassin User's Guide* to assist in future development and enhancement of software documentation, products, and customer service.

The following are trademarks of WDA System Software:

Assassin	StoragePlus	"WDA System Software Helps Manage OpenVMS"
FilesPlus	PagerPlus	

The following are trademarks of Digital Equipment Corporation:

DEC	VAX	All-In-1
VAX/VMS	OpenVMS	VMSinstal
VAXcluster	DECnet	VMSmail

HOW TO ORDER ADDITIONAL PRODUCTS AND DOCUMENTATION

To order additional copies of Assassin, the *Assassin User's Guide*, Assassin literature,or information on other products, contact Networking Dynamics Corporation at the following address.

U.S.A. & INTERNATIONAL

Networking Dynamics Corporation
101 N. Garden Ave. Ste. 220
Clearwater FL. 33755 USA

Phone (727) 446-4511
Fax (727) 446-4252

E-Mail sales@networkingdynamics.com
Web: www.networkingdynamics.com

Contents

DCL Commands

A Installation And Deinstallation

B Miscellaneous Topics

C Assassin Configuration Files

Glossary

Examples

Figures

Tables

Preface

This manual describes how to use the Assassin system software product designed for OpenVMS Systems.

Intended Audience

This document is intended for system managers of all levels of experience.

Document Structure

This manual is divided into **ten** chapters and **three** appendices. It also includes a section describing Assassin's DCL command, a glossary, an index and a section containing important forms.

Chapter 1 introduces Assassin and provides a summary of its capabilities.

Chapter 2 provides information to help you evaluate Assassin.

Chapter 3 describes what Assassin can be configured to do when it identifies an idle process.

Chapter 4 describes how to use Assassin's General Parameter configuration screen.

Chapter 5 provides information that will help you prepare a strategy on to create Exception Records to manage processes on your system.

Chapter 6 describes how you can tell Assassin about special processes on your system so Assassin will not take actions on them. This chapter describes how you can set Assassin to manage processes according to your specifications.

Chapter 7 provides examples of valid Exception Records.

Chapter 8 describes how to start, stop and view the Assassin Monitor Process.

Chapter 9 describes Assassin's Logfiles and their contents.

Chapter 10 describes messages sent by Assassin.

DCL Commands Section describes the Assassin DCL command and its qualifiers.

Appendix A describes how to quickly and easily install Assassin on your system and includes the output from a sample installation.

Appendix B provides additional information about various features and capabilities of Assassin and information for the more advanced Assassin user.

Appendix C explains how to use multiple Configuration Files for specific nodes, days of the week, and times of the day.

The **Glossary** describes terms that are important to understand in order to fully utilize Assassin and the *Assassin User's Guide.*

The **Index** describes where to find issues and topics of particular interest.

1 Introduction

1.1 Introduction

Assassin is an Idle Process Management (IPM) software product that automatically performs predefined actions on processes that are idle, inactive or meet special conditions.

Processes that are idle continue to use numerous resources and prevent other user processes from taking advantage of the VMS operating system's ability to share limited resources. This can seriously impact system performance and place system security at risk. These idle processes can also block other users from accessing VMS or important software application programs by reserving limited license slots.

With Assassin, you can specify:

- What action Assassin will take when a process is idle.
- The level of activity that determines when a process is idle.
- Which processes get special treatment or are exempt.
- Where and how messages are sent when an action is performed.
- The messages that the user receives when an action is taken.
- Who on the systems staff is notified when Assassin takes an action.
- Which terminals are to be "locked" after the user logged in at the terminal has been idle for too much time.
- Which processes will be terminated because the user has been logged in to the system for too long a period of time.
- Which Screen Saver Image is displayed on an idle terminal.
- Different configurations for different nodes at different times of the day and days of the week.

Assassin's configuration component allows you to tailor Assassin to meet the special needs of your organization, system, day of week and time of day.

Assassin's monitoring component periodically gathers information about the processes on your system and decides if an action is required. If a process is consuming CPU time, generating I/Os or pagefaults above the thresholds you specify, then the process is perceived as being busy. Otherwise, it is considered idle. You can define how much resource consumption is required for a process to be considered as busy so that users are measured according to your definition of idle.

Once Assassin is configured as you desire, simply start Assassin's monitor and forget about idle processes on your OpenVMS system forever.

The remainder of this manual describes how Assassin operates and how you may configure Assassin to meet the specific needs of you and your organization.

1.2 Privileges Required To Use Assassin

System Managers must have WORLD, OPER, SHARE, PRMMBX and SYSNAM privileges enabled to operate Assassin and utilize all of its capabilities.

Users utilizing only the /CHECK_USAGE qualifier do not need to have any special privileges. Users do need to have read access to the device and directory containing the Assassin Product Files.

2 How To Easily Evaluate Assassin

2.1 Introduction

This section describes a method you may find useful when evaluating the behavior of Assassin in either a pre-purchase demonstration or when your environment changes, perhaps due to a new application coming on-line or greater user volume.

2.2 Review The *Assassin User's Guide*

This may seem like either an obvious step or a waste of your precious time. However, experience shows that most System Managers who at least skim through this manual will understand Assassin's basic capabilities and gain insights when deciding how to test and configure Assassin.

We have not included any extra material in this manual just to increase its size; most of this manual is as useful to the System Manager testing Assassin for the first time as it is to the experienced Assassin user.

2.3 Choose Safe Default General Parameters

We believe that Assassin's default "out-of-the-box" settings are completely safe and should be suitable for initial testing purposes.

Customers who install Assassin using the default answers to the questions asked by the Product Installation procedure have no trouble testing Assassin. These customers operate Assassin with default, factory-installed, values which can do no harm to your system or applications.

Please use the default Action of "N" ("Notify Operator") to provide notification of idle processes to the operator until you are sure of the characteristics of your system and the corresponding needs of your site.

2.4 The Easiest Way To Test Assassin

The easiest way to test Assassin on your system requires that you perform the following:

1. Install Assassin from the distribution media as described in Appendix A.
2. Issue the DCL Command "ASSASSIN" to start Assassin and then review the General Parameters. **Do not make any changes to this screen until you are familiar with the product and how such modifications will affect Assassin's activity on your system**.
3. Issue the DCL Command "ASSASSIN/MONITOR" and Assassin will start monitoring your system. Assassin will collect data about your system's users.

4. After five minutes, Assassin will recheck your system's users and tell you if any of these users are idle. After a few cycles, you will become aware of which users need to be specially configured into Assassin's Exception Record Screens.
5. Review the *Assassin User's Guide* in more detail to determine the best way in which to amend Assassin's default configuration now that you have viewed Assassin's operation and have a greater understanding of its functionality. At this stage, you will know what level of messaging you would like and how to exempt some of your system users or processes.
6. If you have further questions about Assassin's functionality or operation, call your Assassin distributor, of in the United States call WDA System Software, and as for a "Walk Thru" presentation. This special telephone tutorial will assist you in immediately understanding Assassin and its functionality. Many prospects will be fully configured by the end of this presentation and can utilize the remaining time during the evaluation period actually utilizing and enjoying the benefits of Assassin.

3 Actions And Cycles

3.1 Overview

When Assassin determines that a process is idle for a specified period of time, it will perform the desired Action on that process. You tell Assassin "which" Action to perform when you modify the Default Action Code. You tell Assassin "when" to perform the Action when you modify the Cycle Counts. The default Cycle Length is five minutes long although you can change the length of a cycle using the General Parameters screen. In addition, you can take some actions on processes regardless of the level of activity of the process.

You can then override the Default Action and Cycle Counts by creating Exception Records for the desired processes.

3.2 Actions Consist Of Subactions

Each Action has up to three Subactions. For example, for the Terminate Process With Warning Action (Action Code "T"), the First and Second Subactions send warning messages to the idle process and the Third Subaction sends a message and then terminates the process.

The specific Subactions performed depends upon the Action Code selected for specific processes as described below.

The First, Second and Third Subactions are performed after the process has been idle for the number of cycles defined by Cycle Count One, Two and Three, respectively. Whether all three Subactions are actually performed depends on the Action Code specified, the associated Cycle Counts controlling the idle process, and the ongoing resource consumption of the process.

3.3 Action Code Values

This section describes the different actions Assassin can perform when it determines that a process has been idle too long.

Action B - Blindly Terminate

This action will cause the process to be terminated without sending a warning message or clearing the user's terminal screen. This action is provided for cases where a idle process is to be terminated while remaining completely invisible to the user. For example, you could use this action to terminate an idle subprocess that the user may not be aware of without visually disrupting the user process.

First Action:	Terminate process
Second Action:	No action performed
Third Action:	No action performed

Action C - Custom Command

This action will cause a customizable DCL Command File to be executed when Assassin encounters an idle process. The name of the command file is ASSASSIN_CUSTOM.COM and it is located in that the Product Directory (the same device and directory as the other Assassin Product Files). When this file is executed it receives the Process ID (PID) of the idle process in parameter P1 so that the command procedure can take some site-specific or customized action upon the idle process. For example, you may want to use this action if the user is running a database that needs to have some specific procedure performed before the process is terminated.

You must modify the command file to perform the desired functionality. If you want the user process to be terminated, then the command file must do so (using the STOP/ID DCL Command). When Assassin is installed on your system, a file named ASSASSIN_CUSTOM.TEMPLATE is created (in the Product Directory) which must be copied to ASSASSIN_CUSTOM.COM. The template file is contains some helpful documentation and provides some examples you may want to consider using.

Please refer to the DCL Dictionary, the Guide to Creating Modular Command Procedures or the VMS User's Guide for more information about creating and modifying Command Procedures. These manuals are part of the VMS Extended Documentation Set.

First Action:	Send Message One
Second Action:	Send Message Two
Third Action:	Send Message Three and execute custom procedure

Action E - Exempt Process

This is the "do-nothing" Action. When Assassin encounters an idle process with this Action Code, it will not take any action on it.

There are no Subactions and Cycle Counts are not appropriate. Therefore, any cycle numbers listed for an exempt process will be ignored by Assassin. It is most appropriate to leave the Cycle Counts set to zero ("0") for exempt processes.

Action F - Freeze Process

This action will cause the idle process to be suspended.

First Action:	Send Message One
Second Action:	Send Message Two
Third Action:	Send Message Three and suspend process

Action I - Immediately Termination

This action will cause the idle process to be terminated immediately **whether it is idle or not**. No warning is sent to the process but the user's terminal screen will be cleared. This Action Code cannot be specified as Assassin's Default Action code. **Be careful when using this action code with exception records containing wildcards.**

First Action:	Terminate process immediately
Second Action:	No action performed
Third Action:	No action performed

Action L - Lock Terminal

This action will cause the terminal device of the idle process to be locked**.** The user of the terminal must supply the correct VMS password to unlock the terminal and continue working. If the user required two passwords to log into the system originally, then Assassin will require both passwords to unlock the terminal. The process will perform no work while the terminal is locked. You may be provide a value for the Third Action Cycle Count if the process should be terminated when it has not been unlocked after a period of time. Please refer to **Appendix B** for more information.

First Action:	Send Message One
Second Action:	Send Message Two and Lock the terminal.
Third Action:	Terminate the process

Action N - Notify Operator

This action will cause an idle process notification message to be sent to the designated Operator. You designate the Operator on the Operator Notification Screen. No warning is sent to the idle process and no action is taken on the process.

It is highly recommended that this action be specified when testing Assassin in a new environment or when changes are made to an existing environment. This will avoid disruption to your system caused by the inappropriate termination of an important process.

First Action:	Send Notification To Operator
Second Action:	No action performed
Third Action:	No action performed

Action S - Silently Terminate Process

This action will cause the idle process to be terminated without any previous warning messages being be sent. Warning message number three will be sent just prior to termination.

First Action:	Send Message Number Three and terminate process
Second Action:	No action performed
Third Action:	No action performed

Action T - Terminate Process

This action will cause the idle process to be terminated with one or two warning messages (you configure the number of messages) prior to termination. Once message three has been sent, the process will be terminated regardless of any action the user takes.

First Action:	Send Message One
Second Action:	Send Message Two
Third Action:	Send Message Three and terminate process

Action W - Warn Process Only

This action will cause warning messages one and two to be sent to the idle process. The process will not receive message three and will not be terminated despite what message one and two may indicate.

First Action:	Send Message One
Second Action:	Send Message Two
Third Action:	No action performed

3.4 Cycle Counts and Cycle Length

Actions performed by Assassin are performed at the appropriate Cycle Count. The Cycle Count is equal to the number of Assassin passes on a process since it was identified as idle.

Example 3-1 How Cycle Length And Cycle Counts Interact

Assume that you have set the Cycle Length to five minutes, Cycle Count One to six, Cycle Count Two to seven and Cycle Count Three to eight. Assassin will then perform the First Action (on an idle process) after thirty minutes (6 cycles x 5 minutes = 30 minutes), the Second Action five minutes (7 cycles x 5 minutes = 35 minutes) later, and the Third Action five minutes after the Second Action or forty minutes (8 cycles x 5 minutes = 40 minutes) after being identified as being idle.

If the process becomes busy again at any time between cycles, then its "idle counter" is reset to zero and Assassin will not act on the process until it becomes idle again and reaches the appropriate cycle count threshold.

For further insight on how to select your Cycle Length, please refer to **Section B.8**.

4 General Parameter Configuration Options

4.1 Introduction

This section describes how you can set up Assassin's General Parameters. These parameters control the basic operation of the product.

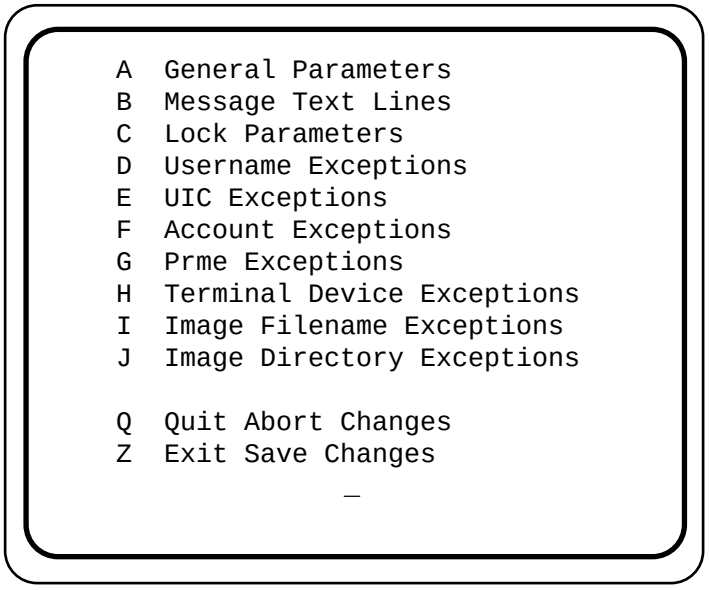
4.2 Main Menu/About Screens

To access Assassin's Main Menu, issue the command:

```
$ Assassin
```

Assassin will present the following menu:

Figure 4-1 Main Menu



The Assassin Main Menu presents a list of configuration screens which you can access to configure Assassin according to your specific system needs. The configuration screens are presented and described below.

Configuration screens are accessed by entering the letter corresponding to the desired screen.

You may press the F7 function key at any time to display Assassin's ABOUT screen. These informational screens will provide important information about the current version of Assassin software you are running. It will also communicate information about where to contact WDA System Software should you have a need for technical service or wish to purchase additional copies of Assassin.

4.3 General Parameters Screen

Figure 4-2 General Parameters Screen

```
Default Action        N Notify    Cycle Counts Local Dialup Remote
Cycle Length          5 Minutes     Action 1       6      6      6
Batch Jobs Exempt?    Yes           Action 2       7      7      7
Detached Jobs Exempt? Yes           Action 3       8      8      8

Idle Threshold Values              Notification Style
  CPU Time     0 Buffered I/Os  0     Via Broadcast   3 Verbose
  Direct I/Os  0 Page Faults    0     Via OPCOM       3 Verbose
                                      Via Mail        3 Verbose
Miscellaneous
  Broadcast Address   SYSTEM
  Mail Address        SYSTEM
  Mail DCL Command    MAIL
  Timestamp User      SYSTEM
  Timestamp Freq      180 Minutes     Screen Saver     0
  Process Name        Assassin_Monitor  Purge Memory?   No
  MaxConnect Time       0 Cycles      Logfile Content 0 Operator
```

4.3.1 Data Fields

Default Action

This is the field where you specify Assassin's Default Action. Assassin will perform the specified action on all processes that have been idle longer than the desired time limit. The desired time limit is calculated by multiplying the Cycle Length by the Cycle Count for a specific Action. Only processes that have been listed as exceptions will receive an action different from the default. **The Default Action field cannot be left blank. A value is required for this field.** The installation process sets the Action to "N" ("Notify Operator") as the default. If no other action is specified then the Notification Action is used.

The Immediate Terminate ("I") Action is not allowed as a default action.

Valid entries for this field are shown in the following table:

Default: "N" - Notify Operator

Example: An entry of "T" will cause all idle processes to be terminated.

Table 4-1 Valid Default Action Codes

Value	Meaning
B	Blindly Terminate
C	Custom Command
E	Exempt Process
F	Freeze (Suspend) Process
L	Lock Terminal
N	Notify Operator (Default Value)
S	Silently Terminate Process
T	Terminate Process
W	Warn User Process

Cycle Length

This field defines the length in minutes of Assassin's process monitoring cycle. The installation process sets this field to five minutes as the default Cycle Length. With this default setting, once Assassin is started, it will wake up and check for idle processes every five minutes after the completion of its last cycle.

Default: "5" - Five Minutes

Example: An entry of "15" will cause Assassin to scan your system every fifteen minutes.

Batch Jobs Exempt?

This field controls whether or not Batch Jobs are always exempt on your system. A Batch Job is defined as a job (process) running from a Batch Queue. Set this field to "Y" ("Yes") to make Batch Jobs always exempt or "N" ("No") to allow Assassin to process them according to its other rules.

Default: "N" - Batch Jobs Are Not "Always Exempt"

Example: An entry of "Y" will automatically exempt all Batch Jobs.

Detached Jobs Exempt?

This field controls whether or not Detached Jobs are always exempt on your system. A Detached Job is defined as a job (process) running without a user terminal. Set this field to "Y" ("Yes") to make Detached Jobs always exempt or "N" ("No") to allow Assassin to process them according to its other rules.

Default: "N" - Detached Jobs Are Not "Always Exempt"

Example: An entry of "Y" will automatically exempt all Detached Jobs.

General Parameter Configuration Options
4.3.1 Data Fields

Cycle Count - First Action

These fields store the number of cycles for which a process must be idle before the First Action will be performed for Local, Dialup and Remote Processes. Entries can range from "0" through "9999". If the entry is zero ("0"), then Assassin will not perform the First Action called for by the Action Code.

Default: "6" - Perform the first action after six cycles.

Example: An entry of "10" will cause Assassin to perform the First Action after a process has been idle for ten cycles.

Cycle Count - Second Action

These fields store the number of cycles for which a process must be idle before the Second Action will be performed for Local, Dialup and Remote Processes. Entries can range from "0" through "9999" and must be greater than the entry in the Cycle Count - First Action field of the same process type. If the entry is zero ("0"), then Assassin will not perform the Second Action called for by the Action Code.

Default: "7" - Perform the first action after seven cycles.

Example: An entry of "12" will cause Assassin to perform the First Action after a process has been idle for twelve cycles.

Cycle Count - Third Action

These fields store the number of cycles for which a process must be idle before the Third Action will be performed for Local, Dialup and Remote Processes. Entries can range from "0" through "9999" and must be greater than the entry in the Cycle Count - Second Action field of the same process type. If the entry is zero ("0"), then Assassin will not perform the Third Action called for by the Action Code.

Default: "8" - Perform the first action after eight cycles.

Example: An entry of "14" will cause Assassin to perform the First Action after a process has been idle for fourteen cycles.

CPU Time Threshold

This field defines the amount of CPU time (in "ticks") that a process must consume to be considered busy. If this field is set to zero ("0"), the default value, then a process consuming any CPU time during a cycle will be perceived as being busy. If your users or applications tend to use small amounts of CPU time without "being busy", then you should set this value slightly above that usage level so that Assassin can act on those processes when they are idle. See **Section B.5** of this manual for more information about determining values for this field.

Default: "0" - No CPU Time Consumed

Example: An entry of "50" will cause Assassin to mark processes as busy when they consume more than fifty-hundredths of a CPU Second.

Direct I/O Threshold

This field defines the number of Direct I/Os that a process must generate to be considered busy. If this field is set to zero ("0"), the default value, then a process generating any Direct I/Os during a cycle will be perceived as being busy. If your users or applications tend to generate a small number of Direct I/Os without "being busy", then you should set this value slightly above that usage level so that Assassin can act on those processes when they are idle. See **Section B.5** of this manual for more information about determining values for this field.

Default: "0" - No Direct I/Os Generated

Example: An entry of "10" will cause Assassin to mark processes as busy when they generate more than ten Direct I/O Operations.

Buffered I/O Threshold

This field defines the number of Buffered I/Os that a process must generate to be considered busy. If this field is set to zero ("0"), the default value, then a process generating any Buffered I/Os during a cycle will be perceived as being busy. If your users or applications tend to generate a small number of Buffered I/Os without "being busy", then you should set this value slightly above that usage level so that Assassin can act on those processes when they are idle. See **Section B.5** of this manual for more information about determining values for this field.

Default: "0" - No Buffered I/Os Generated

Example: An entry of "20" will cause Assassin to mark processes as busy when they generate more than twenty Buffered I/O Operations.

General Parameter Configuration Options
4.3.1 Data Fields

Page Fault Threshold

This field defines the number of Page Faults that a process must generate to be considered busy. If this field is set to zero ("0"), the default value, then a process generating any Page Faults during a cycle will be perceived as being busy. If your users or applications tend to generate a small number of Page Faults without "being busy", then you should set this value slightly above that usage level so that Assassin can act on those processes when they are idle. See **Section B.5** of this manual for more information about determining values for this field.

Default: "0" - No Pagefaults Generated

Example: An entry of "15" will cause Assassin to mark processes as busy when they generate more than fifteen pagefaults.

Notification Style Via Broadcast

This field defines the format of the Notification Message sent to the desired operator via broadcast services. Valid entries for this field are 0, 1, 2 or 3. If the value is zero ("0"), then no message is sent. Otherwise, the entry determines the style or length of messages sent. Message Style 1 is brief, Message Style 2 includes additional information and Message Style 3 includes the most information. Please refer to **Chapter 10** for examples of the different Message Styles.

Default: "1" - Broadcast messages will contain the minimal information

Example: An entry of "3" will cause Assassin to include the most information in its Operator Notification Messages.

Notification Style Via OPCOM

This field defines the format of the Notification Message sent to the desired operator via the OPCOM process. Valid entries for this field are 0, 1, 2 or 3. If the value is zero ("0"), then no message is sent. Otherwise, the entry determines the style or length of messages sent. Message Style 1 is brief, Message Style 2 includes additional information and Message Style 3 includes the most information. Please refer to **Chapter 10** for examples of the different Message Styles.

Default: "1" - OPCOM messages will contain the minimal information

Example: An entry of "3" will cause Assassin to include the most information in its Operator Notification Messages.

Notification Style Via Mail

This field defines the format of the Notification Message sent to the desired operator via Mail. Valid entries for this field are 0, 1, 2 or 3. If the value is zero ("0"), the no message is sent. Otherwise, the entry determines the style or length of messages sent. Message Style 1 is brief, Message Style 2 includes additional information and Message Style 3 includes the most information. Please refer to **Chapter 10** for examples of the different Message Styles.

Default: "3" - Mail messages will contain the most information

Example: An entry of "0" will cause Assassin not to send any Operator Notification Messages by Mail.

Broadcast Address

This field contains the username of the recipient(s) of Assassin's Broadcast Messages. You may enter a single username or a list of usernames separated by commas.

Default: Username of first user to start Assassin after installation

Example: An entry of "SMITH,OPCOM,JONES" will cause Assassin to send Broadcast Messages to users Smith and Jones and to all Operator Consoles.

Mail Address

This field contains the username of the recipient(s) of Assassin's Mail Messages. Any valid mail address for the mail software you choose is acceptable. You may enter a single username, a list of usernames separated by commas.

Default: Username of first user to start Assassin after installation

Example: An entry of "JONES,VAXA::SMITH,@MGRLIST" will cause Assassin to send Mail Messages to user Jones, user Smith on Node VAXA, and to all users in the MGRLIST distribution list.

General Parameter Configuration Options
4.3.1 Data Fields

Mail DCL Command

This field contains the DCL command (without the dollar sign) that will be used to send Notification Messages via Mail. The default value of "MAIL" will cause Assassin to use VMSmail. Please refer to **Section B.6** of this manual for information about sending mail via other mail sending agents.

Default: "MAIL" - Use VMSmail facility

Example: An entry of "@SYS$MANAGER:MAILSENDER" will cause Assassin to use MAILSENDER.COM in the SYS$MANAGER directory to send mail messages.

Timestamp User

This field contains the username of the recipient(s) of Timestamp Messages. The Assassin Monitor Process will regularly send timestamp messages to the user specified based on the frequency specified in the Timestamp Freq field above. Enter "OPCOM" to send messages to the Operator Console via OPCOM. You may enter a single username or a list of usernames separated by commas.

Default: Username of first user to start Assassin after installation

Example: An entry of "SYSTEM,OPCOM" will cause Assassin to send Timestamp Messages to user System and to all Operator Consoles.

Timestamp Freq

This field specifies the frequency, in minutes, that the Assassin Monitor Process will send Timestamp Messages to the user you specify in the Timestamp User field. If you enter "30", then the Assassin Monitor Process will send a timestamp message every thirty minutes to confirm that is running successfully. Enter zero ("0") to disable the sending of messages.

Default: "180" - Send Timestamp Messages every sixty minutes

Example: An entry of "360" will cause Assassin to send Timestamp Messages every three hours.

Process Name

This field defines the Process Name of the Monitor Detached Process, The Process Name of the Monitor Detached Process will be the value you enter here. You may enter a mixed-case Process Name value.

Default: "Assassin_Monitor" - Assassin will use a Process Name of "Assassin_Monitor.

Example: An entry of "WDA" will cause Assassin to use a Process Name of "WDA".

MaxConnect Time Cycle Count

This field defines the default Maximum Connect Time a process is allowed before it will be logged off. An entry of zero ("0") will disable this feature. The time is expressed as a number of Monitoring Cycles. This field allows you to have processes terminated after they have been using the system for a desired period of time.

Processes will be terminated when they have been logged in longer than the number of cycles specified in this field whether the process is idle or not.

Default: "0" - MaxConnect Time is disabled.

Example: An entry of "10" will cause processes to be terminated after they have been logged in longer than eight cycles. They will be warned at least two cycles and again one cycle before they are terminated.

Screen Saver Image Number

This field allows you to specify the Screen Saver Image that will be used by the Assassin Screen Saver function. An entry of zero ("0") will disable this feature. Assassin will write the Screen Saver Image you specify here on idle terminal screens. Please refer to **Appendix C** for more information

Default: "0" - Screen Saver is disabled.

Example: An entry of "3" will cause Assassin to use Screen Saver Image 3.

General Parameter Configuration Options
4.3.1 Data Fields

Purge Memory

This field determines whether or not the Monitor Process will purge its working set when it is hibernating. Enter "Y" if you want the Monitor Process to purge its Working Set or "N" if you do not. If the Working Set is purged, more memory will be available on your system, but the Monitor Process will generate more pagefaults when it becomes active as it reloads its ongoing data into its working set.

Default: "No" - Assassin will not purge its Working Set.

Example: An entry of "Y" will cause Assassin to purge its Working Set.

Logfile Contents

This field defines what messages Assassin writes to the Actions Logfile. The value entered here must be either "A" ("All Messages") to have all messages written, "O"("Operator Notification Messages") to write messages when the Operator is notified of Actions taken, or "F" ("Final Action Messages") to write messages only when final subaction are taken. Setting the value to "F" can significantly reduce the number of messages written to the Logfile and will reduce the disk space that the logfile occupies.

Default: "O" - Write Operator Notification messages to the Logfile

Example: An entry of "A" will cause Assassin to write all messages to the Actions Logfile.

4.4 Message Text Lines Screen

Figure 4-3 Message Text Lines Screen

```
Message Text
  First Action
    Please use the VAX or log off the system. When you are logged in and
    not active, you still consume resources and slow down the VAX system.
  Second Action
    Your process is still idle. Please log off the system so others may
    benefit from resources and applications you currently have reserved.
  Third Action
    Your process has been idle too long. This threatens system security
    and system performance. Your process will therefore be terminated.

  MaxConnect Time Warning
    You are nearing your Connect Time Limit and will be logged out soon.
  MaxConnect Time Final
    You have exceeded your Connect Time Limit and will be logged out now.
```

4.4.1 Data Fields

The following fields appear on the Message Text Lines Screen:

First Action Message Text

These fields contain the two lines of text sent to a process when the First Action is performed. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

Second Action Message Text

These fields contain the two lines of text sent to a process when the Second Action is performed. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

Third Action Message Text

These fields contain the two lines of text sent to a process when the Third Action is performed. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

MaxConnect Time Warning

This field contains the line of text sent to a process, as a warning message, about two cycles before it will be terminated because it has been logged in too long. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

MaxConnect Time Final

This field contains the line of text sent to a process, as a final message, just before it is to be terminated because it has been logged in too long. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

4.4.2 Choose Message Text Lines Wisely

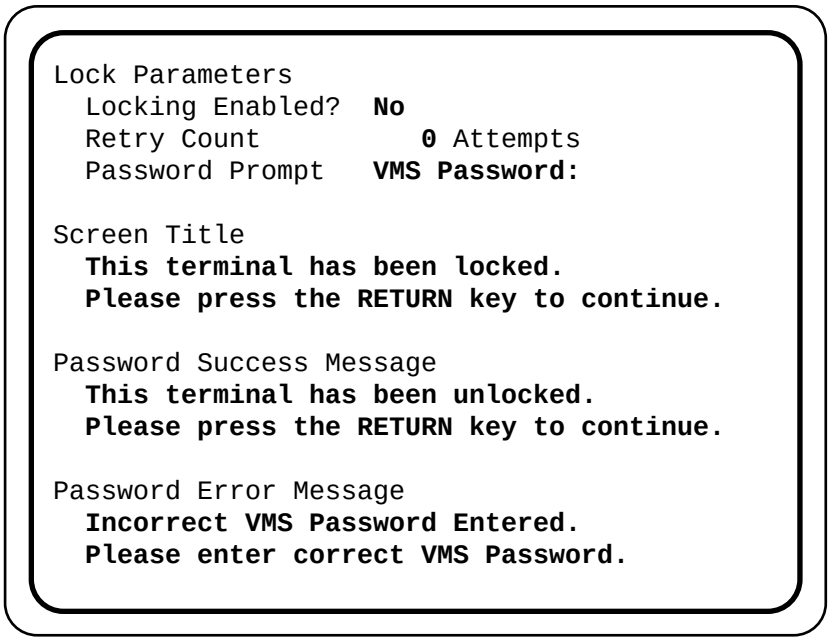
These messages are only the default messages and can be changed as you desire to meet the communication needs and tone of your site.

Significant time and care has been provided in writing, researching and testing the default message text on system users at customer sites. These messages have proven to be the most informative, educational and congenial.

Caution is recommended when customizing messages for use at your site. Remember that end users do not always understand the benefits of terminating idle processes and may react strongly if terminated unexpectedly with a message like "Ha, Ha, You're Dead". By educating your system users and warning them in advance you can avoid static and rebellion from users and managers.

4.5 Lock Parameters Screen

Figure 4-4 Lock Parameters Screen



4.5.1 Data Fields

The following fields appear on the Lock Parameters Screen:

Locking Enabled?

This field whether Assassin's Terminal Locking capabilities are enabled or disabled for your system. Enter "Yes" to enable or "No" to disable Terminal Locking at your site.

Default: "No" - Assassin Terminal Locking is disabled.

Retry Count

This field defines how many times the user is prompted for the correct password to unlock the terminal before the process is terminated. Enter zero ("0") to allow an unlimited number of attempts.

Default: "0" - Users have an unlimited number of attempts to provide the correct VMS Password.

Example: An entry of "5" would terminate the process when a user has provided an incorrect password five times in a row.

General Parameter Configuration Options
4.5.1 Data Fields

Password Prompt

This field allows you to specify the prompt Assassin will use when asking the user for their VMS account login password. Assassin requires that the user provide the correct password before unlocking the terminal and will use the text you specify in this field as the prompt.

Default: The default for this field appears in the screen shown above.

Screen Title

These fields contain the two lines of text sent to a process when Assassin initially locks the user's terminal. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

Password Success Message

These fields contain the two lines of text sent to a process when the user has successfully unlocks the terminal. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

You could use these fields to convey any necessary information about how to "refresh" the user's terminal screen after it is unlocked.

Default: The default for this field appears in the screen shown above.

Example: If the user application refreshes the screen when a Control-W is pressed, then you could enter the following message in this field: "Press Control-W to refresh screen".

Password Error Message

These fields contain the two lines of text sent to a process when the user has provided an invalid or incorrect password for the locked terminal. The text in these fields can be edited for custom messages and will be sent exactly as it is entered.

Default: The default for this field appears in the screen shown above.

5 Configuring Exceptions

5.1 Overview

The Assassin Configuration File specifies a Default Action that will be performed when it encounters an idle process. This Default Action should be set appropriately for the majority of the processes on your system. However, most systems will also have a need to treat certain processes in a special way. For those processes, Assassin allows you to create Exception Records. An exception record is a process that is listed in one of Assassin's exception screens and specifies an Action or Cycle Count different from that specified in the General Parameters Screen.

When Assassin is determining what action to take on an idle process, it first checks through all of the Exception Records to see if the idle process matches one of the specified exception criteria. If there is no match, then Assassin uses the Default Action and Cycle Counts to determine how to handle the idle process. If there is a match, then Assassin uses the Action Code and Cycle Counts from the Exception Record instead of the Default Actions and Cycle Counts.

You may create up to 128 different Configuration Records using each of the Exception Record maintenance screens. To enter additional information once a screen is full, press the Next Screen or Control-N keys. To return to a previous screen of information, press the Prev Screen or Control-P keys. Press the DO or PF4 keys when you have finished entering Exception Record information.

5.2 Certain Process Are Always Exempt

There are certain processes that are important to the correct and secure operation of VMS and popular software applications. Occasionally, some of these processes will be idle. Because terminating these processes could create undesirable situations, Assassin is designed to **always** exempt these processes.

The following lists describe processes that are always exempt and therefore will not be affected by Assassin. You do not need to explicitly list any of the following processes as exceptions.

1. Any DECwindows Process whose image name begins with the characters "DECW$".
2. Any process whose UIC group number is one (i.e. [1,*]).
3. Any SYSTEM username process (Username is SYSTEM).
4. Any process with a subprocess. (Parent processes that are idle and have subprocesses will be acted upon only after all subprocesses have been terminated or logged out.)
5. Any process listed in the following table.

Table 5-1 Specific Processes That Are Always Exempt

Type Of Process	Process Name
VMS Operating System	AUDIT_SERVER CACHE_SERVER CLUSTER_SERVER CONFIGURE ERRFMT EVL ICPACP JOB_CONTROL LATACP NETACP OPCOM REMACP SMISERVER SWAPPER SYMBIONT_nnnn TP_SERVER SPM_TUNE VPA_DC
Pathworks (PCSA)	LAD$KERNEL NETBIOS PCFS_SERVER PWRK$*
Database Monitors	DBMS_MONITOR RDMS_MONITOR
Other Products	UCX$*

5.3 Creating Exception Records

Exception Records are created by listing a process as an exception using one of seven exception criteria available in Assassin. You can define Exceptions for processes via Username, UIC, Account (from the UAF record's Account field), Process Name, Terminal Device, Image Filename and Image Directory. The Image Filename and Image Directory for a Process are based on the program the process is executing when Assassin gathers data for that process. The other process criteria typically are permanently defined at the time the process is created.

5.4 Exception Records Have A Precedence

You can create Exception Records so that a user's process can be covered by multiple exception criteria simultaneously. However, these exception criteria will follow a precedence where one criteria will override another criteria. When configuring for many different process situations, it is important to understand the precedence of Exception Records.

Example 5-1 Exception Record Precedence

For example, you may want users that use the system through a TX terminal device to be exempt. But you have users in UIC Group 231 that need to be warned, even if they use a TX terminal device. To set this up, you create a

Terminal Device Exception Record with "TX*" as the Terminal and "E" (Exempt) as the Action Code. Then you create a UIC Exception Record where the UIC value is [231,*] and the Action Code is "W" (Warn).

The criteria listed below are listed in precedence order. Those criteria listed on the bottom of the list have a higher precedence over those listed on the top of the list. The order of precedence, from lowest to highest, is as follows:

1. Terminal Server/Remote Terminal/Network Node (lowest precedence)
2. Terminal Name
3. UAF Record Account
4. UIC Code
5. Username
6. Process Name
7. Image Directory
8. Image Filename (highest precedence)

Also, specific Exception Records listed lower in an exception listing will always take precedence over an entry of the same type listed in the same exception list. If you have three Username Exception Records and a process can be identified by more than one of the three Username Exception Record entries, then the entry that appears lowest on the list will take precedence over any entry listed above it.

Example 5-2 Exception Record Precedence

Username	Action	Cyc1	Cyc2	Cyc3
FOUTS	E	0	0	0
FR*	W	1	2	0
FRED	T	1	3	5
FRANKENSTEIN	S	2	0	0

User FOUTS is exempt, therefore no action will be taken on this process.
User FRED will be warned after one and three idle cycles and then terminated after five idle cycles.
User FRANK will be warned after one and two idle cycles.
User FRANKENSTEIN will be terminated silently after two idle cycles.

5.5 Exception Records Accept Wildcard Values

Wildcard values can be used when creating any Exception Record except for Image Directory Exceptions. Valid exception entries both with and without wildcards are shown below. Note that the wildcard character ("*") can appear at the beginning, middle, one end, or both ends of the value entered.

Using wildcards can sometimes reduce the number of Exception Record entries required to manage a group of users. However, be careful when using wildcards as you may inadvertently include users that you did not intend to include. Whenever possible, use broader exception types like UICs, Accounts, and others to manage groups of users.

Example 5-3 Sample Exception Records And Wildcard Values

Record Type	Without Wildcard	With Wildcard
Terminal Name	LTA1234: TXA2: RTA23: SERVER/PORT_1 REMNOD::JONES	LTA* TX*2 RTA1*: SRVR_A/* */PORT_7 *::SMITH
UAF Record Account	FINANCE PUBS	TEST* PUB*
UIC Code	[123,456] [654,321]	[123,*] [456,*]
Username	JONES SMITH	PAYROLL* MAIL$*
Process Name	DBSERVER JOHN_LTA12	ORACLE* INFX* *$SERVER
Image Filename	BACKUP CLEANUP LOOKOUT	ANA* DBPOST*
Image Directory	[DATABASE] [PROG.EXE]	Not Allowed

5.6 Exception Record Screens

The remainder of this chapter describes Assassin's Exception Record Screens. You create Exceptions using these screens so that Assassin will perform according to your precise needs.

Each screen is described individually with some example information for your reference. Fields that appear on all screens are described in the next section.

5.7 Fields Common To All Exception Record Screens

The following fields appear on each Exception Record Screen and perform the same function on each screen. They are included here for easy reference.

Action Code

This is the field where you specify the Action that Assassin is to take for this exception record. Assassin will perform the chosen action on all processes

affected by this exception record that have been idle longer than the desired time limit. The desired time limit is calculated by multiplying the Cycle Length by the Cycle Count. **This field cannot be left blank for a valid exception record.**

Valid entries for this field are shown in below

Table 5-2 Valid Exception Record Action Codes

Value	Meaning
B	Blindly Terminate
C	Custom Command
E	Exempt Process
F	Freeze (Suspend) Process
I	Immediately Terminate Process
L	Lock Terminal
N	Notify Operator (Default Value)
S	Silently Terminate Process
T	Terminate Process
W	Warn User Process

Cycle Count - First Action

These fields store the number of cycles for which a process affected by this exception record must be idle before the First Action will be performed. Entries can range from "0" through "9999". If the entry is zero ("0"), then Assassin will not perform the First Action called for by the Action Code.

Cycle Count - Second Action

These fields store the number of cycles for which a process affected by this exception record must be idle before the Second Action will be performed. Entries can range from "0" through "9999" and must be greater than the entry in the Cycle Count - First Action field. If the entry is zero ("0"), then Assassin will not perform the Second Action called for by the Action Code.

Cycle Count - Third Action

These fields store the number of cycles for which a process affected by this exception record must be idle before the Third Action will be performed. Entries can range from "0" through "9999" and must be greater than the entry in the Cycle Count - Second Action field. If the entry is zero ("0"), then Assassin will not perform the Third Action called for by the Action Code.

Screen Saver Image Number

This field allows you to specify, for this exception record, the Screen Saver Image that will be used by the Assassin Screen Saver function. An entry of zero ("0") will disable this feature. Assassin will write the Screen Saver Image you specify here on idle terminal screens. Please refer to **Appendix**

C for more information

Default: "0" - Screen Saver is disabled.

Example: An entry of "3" will cause Assassin to use Screen Saver Image 3.

MaxConnect Time Cycle Count

This field defines the Maximum Connect Time for this exception record. A process will be logged off when its connect time exceeds the number of cycles you enter here. An entry of zero ("0") will disable this feature. The time is expressed as a number of Monitoring Cycles. This field allows you to have processes terminated after they have been using the system for a desired period of time.

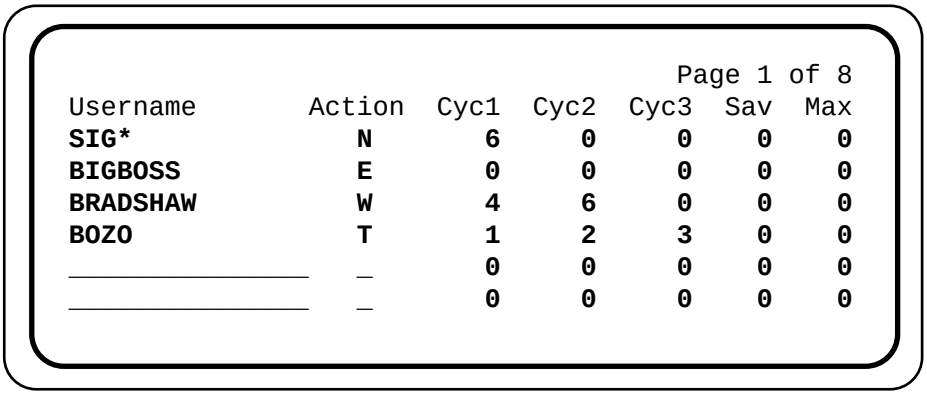
Processes will be terminated when they have been logged in longer than the number of cycles specified in this field whether the process is idle or not.

Default: "0" - MaxConnect Time is disabled.

Example: An entry of "10" will cause processes to be terminated after they have been logged in longer than eight cycles. They will be warned at least two cycles and again one cycle before they are terminated.

5.8 Username Exception Screen

Figure 5-1 Username Exception Screen



```
                                                    Page 1 of 8
Username        Action  Cyc1  Cyc2  Cyc3  Sav  Max
SIG*               N       6     0     0    0    0
BIGBOSS            E       0     0     0    0    0
BRADSHAW           W       4     6     0    0    0
BOZO               T       1     2     3    0    0
________________   _       0     0     0    0    0
________________   _       0     0     0    0    0
```

Username Field

This field contains the username of processes to be granted exceptions via their Username. Usernames can be explicitly stated or may contain wildcard values. Using wildcards can sometimes reduce the number of entries required to manage a group of users. However, be careful when using wildcards as

you may include users that you did not intend to include. Whenever possible, use broader exception types like UICs, Accounts, and others to manage groups of users.

5.9 Sample Username Exception Records

- The first entry shown above would cause Assassin to send an idle notification message to the operator when any username that starts with "SIG" has been idle for thirty minutes (6 cycles x 5 minutes).
- The second entry shown above shows that user BIGBOSS is exempt, therefore Assassin will take no action on this process.
- The third entry shown above will cause one warning message to be sent to user BRADSHAW when he has been idle for twenty minutes (4 cycles x 5 minutes) and a second warning message to be sent when user BRADSHAW has been idle for thirty minutes (6 cycles x 5 minutes).
- The last entry shown above will cause Assassin to terminate user BOZO when that user has been idle for fifteen minutes (3 cycles x 5 minutes). Warnings will be sent after five and ten minutes of inactivity.

5.10 UIC Exception Screen

Figure 5-2 UIC Exception Screen

```
                                                    Page 1 of 8
UIC              Action  Cyc1  Cyc2  Cyc3  Sav  Max
[1234,*]            E       0     0     0    0    0
[367,*]             S       8     0     0    0    0
[55,*]              W       4     8     0    0    0
[*,10]              N       5     0     0    0    0
________________    _       0     0     0    0    0
________________    _       0     0     0    0    0
```

UIC Field

This field contains the UIC code of processes to be granted exceptions via their UIC. UIC codes must be entered in numeric format, including the square brackets, and may contain group and/or member wildcards ("*").

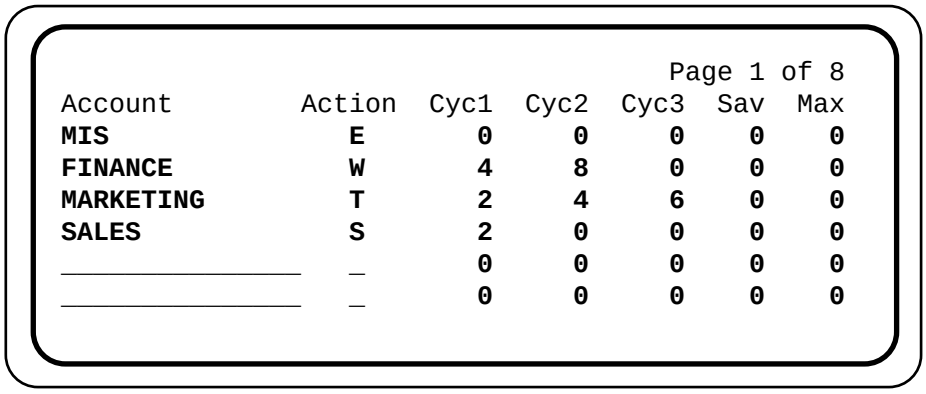
Note: All users in UIC Group 1 are always automatically exempt and need not be entered in this screen. This is because most standard VMS Operating System processes are members of UIC Group 1.

5.11 Sample UIC Exception Records

- The first entry shown above tells Assassin that any user in UIC Group 1234 is exempt, therefore Assassin will take no action on these processes.
- The second entry shown above tells Assassin to terminate users in UIC Group 367 without warning after they have been idle for 40 minutes (8 cycles x 5 minutes).
- The third entry shown above causes Assassin to send warning messages to users in UIC Group 55 after twenty minutes (4 cycles x 5 minutes) and forty minutes (8 cycles x 5 minutes).
- The last entry shown above will cause Assassin to send an idle notification message to the Operator when any user with a UIC member number of 10 has been idle for twenty-five minutes (5 cycles x 5 minutes).

5.12 Account Exception Screen

Figure 5-3 Account Exception Screen



```
                                                    Page 1 of 8
Account          Action  Cyc1  Cyc2  Cyc3  Sav  Max
MIS                 E       0     0     0    0    0
FINANCE             W       4     8     0    0    0
MARKETING           T       2     4     6    0    0
SALES               S       2     0     0    0    0
________________    _       0     0     0    0    0
________________    _       0     0     0    0    0
```

Account Name Field

This field contains the account of processes to be granted exceptions via the Account value as defined in each user's SYSUAF record. Account values can be explicitly stated or may contain wildcard ("*") values.

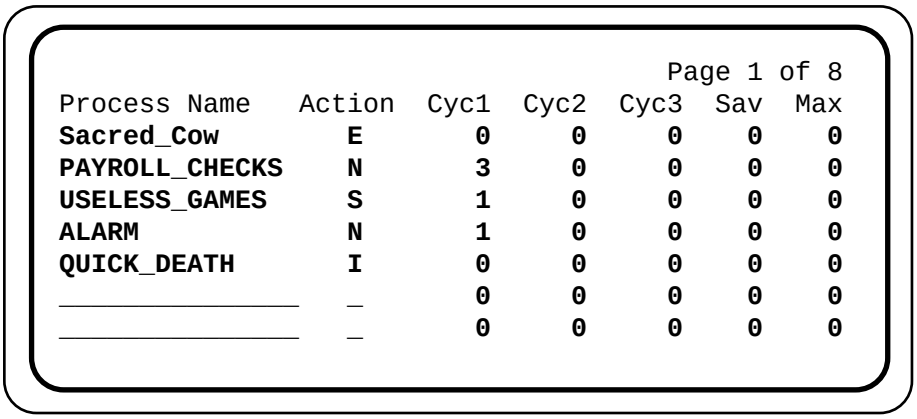
5.13 Sample Account Exception Records

- The first entry shown above tells Assassin that any user in the MIS group will be exempt, therefore Assassin will take no action on these processes.
- The second entry shown above shows that users in the FINANCE group will be warned after they have been idle for twenty minutes (4 cycles x 5 minutes) and forty minutes (8 cycles x 5 minutes).
- The third entry shown above shows that users in the MARKETING group will be warned after ten minutes (2 cycles x 5 minutes) and twenty minutes (4 cycles x 5 minutes) and terminated after being idle for thirty minutes (6 cycles x 5 minutes).

- The last entry shown above shows that SALES users will be terminated without warning after they have been idle for ten minutes (2 cycles x 5 minutes).

5.14 Process Name Exception Screen

Figure 5-4 Process Name Exception Screen



```
                                                    Page 1 of 8
Process Name     Action  Cyc1  Cyc2  Cyc3  Sav  Max
Sacred_Cow          E       0     0     0    0    0
PAYROLL_CHECKS      N       3     0     0    0    0
USELESS_GAMES       S       1     0     0    0    0
ALARM               N       1     0     0    0    0
QUICK_DEATH         I       0     0     0    0    0
________________    _       0     0     0    0    0
________________    _       0     0     0    0    0
```

Process Name Field

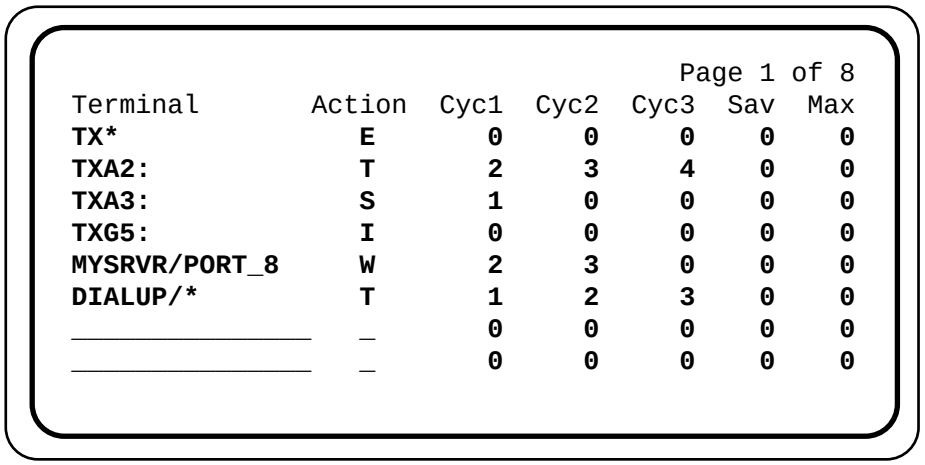
This field contains the process name of processes to be granted exceptions via their Process Name. Process Names can be explicitly stated and may contain wildcard ("*") values. The values entered here are case sensitive and must be entered in the way that they appear on your system. You may issue the DCL command "SHOW SYSTEM" to display the Process Names of the processes on your system.

5.15 Sample Process Name Exception Records

- The first entry above shows that the process named "Sacred Cow" is exempt, therefore Assassin will take no action on this process.
- The second entry above shows that Assassin will send an idle notification message to the Operator when the process named "PAYROLL_CHECKS" has been idle for fifteen minutes (3 cycles x 5 minutes).
- The third entry above shows that any process named "USELESS_GAMES" will be terminated without warning after it has been idle for five minutes (1 cycle x 5 minutes).
- The fourth entry above shows that Assassin will send an idle notification message to the Operator when the process named "ALARM" has been idle for more than five minutes.
- The last entry above shows that the process named "QUICK_DEATH" will be terminated immediately as soon as Assassin finds it. **This process will be terminated even if it is busy.**

5.16 Terminal Exception Screen

Figure 5-5 Terminal Exception Screen



```
                                                    Page 1 of 8
Terminal          Action   Cyc1   Cyc2   Cyc3   Sav   Max
TX*                  E        0      0      0     0     0
TXA2:                T        2      3      4     0     0
TXA3:                S        1      0      0     0     0
TXG5:                I        0      0      0     0     0
MYSRVR/PORT_8        W        2      3      0     0     0
DIALUP/*             T        1      2      3     0     0
________________     _        0      0      0     0     0
________________     _        0      0      0     0     0
```

Terminal Device Field

This field contains the Terminal Identifier of processes to be granted exceptions via their Terminal. Terminal Identifiers can be any of the following:

- Terminal Device Names ("TTA0:")
- LAT Device Names ("LTA123:")
- Terminal Server/Port Names ("MYSRVR/PORT_12")
- X.29 Network Connections ("network.remote_dte")
- DECnet Remote Terminals ("RTA2:")
- DECnet Remote Node::User Connections ("MYNODE::JONES")

Terminal Identifiers can be explicitly stated and may contain wildcard ("*") values. These entries may, although do not need to, include a colon (":") where appropriate.

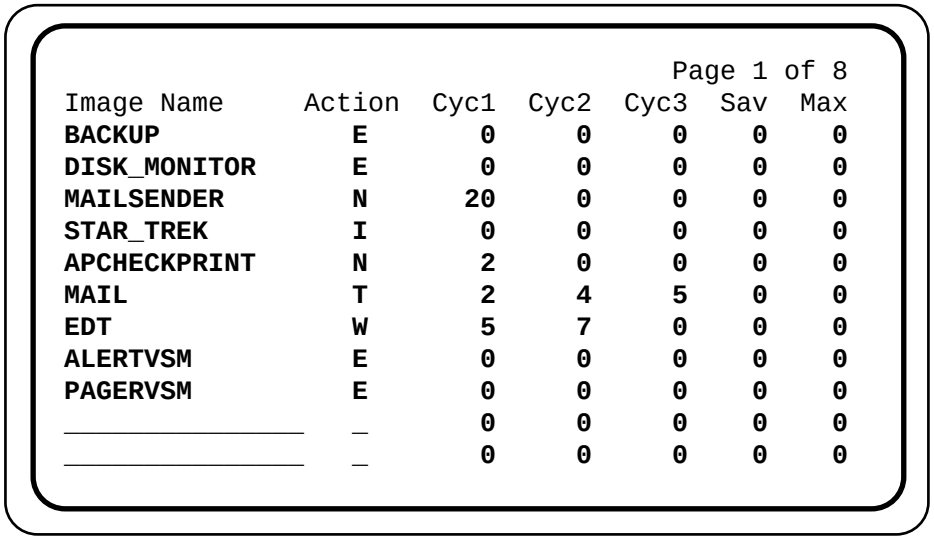
5.17 Sample Terminal Exception Records

- The first entry above shows that all processes using "TX" terminal ports will be exempt, therefore Assassin will take no action on these processes.
- The second entry above shows that processes using terminal port TXA2: will be will be warned after ten minutes (2 cycles x 5 minutes) and fifteen minutes (3 cycles x 5 minutes). These processes will then be terminated after they have been idle for twenty minutes (4 cycles x 5 minutes).

- The third entry shown above will cause Assassin to silently terminate, without warning, any process using terminal port TXA3: after it has been idle for five minutes (1 cycle x 5 minutes).
- The fourth entry shown above will cause Assassin to immediately terminate any process using terminal port TXG5: without warning and without regard to any level of activity by the process.
- The fifth entry shown above will cause Assassin to warn processes connected to the system via PORT_8 on Terminal Server MYSRVR after they have been idle for ten minutes (2 cycles x 5 minutes) and fifteen minutes (3 cycles x 5 minutes).
- The last entry shown above will cause Assassin to terminate any idle process using any port on Terminal Server DIALUP after fifteen minutes (3 cycles x 5 minutes). Warnings will be sent after five (1 cycle x 5 minutes) and ten (2 cycles x 5 minutes) minutes of idle time.

5.18 Image Filename Exception Screen

Figure 5-6 Image Filename Exception Screen



Page 1 of 8

Image Name	Action	Cyc1	Cyc2	Cyc3	Sav	Max
BACKUP	**E**	**0**	**0**	**0**	**0**	**0**
DISK_MONITOR	**E**	**0**	**0**	**0**	**0**	**0**
MAILSENDER	**N**	**20**	**0**	**0**	**0**	**0**
STAR_TREK	**I**	**0**	**0**	**0**	**0**	**0**
APCHECKPRINT	**N**	**2**	**0**	**0**	**0**	**0**
MAIL	**T**	**2**	**4**	**5**	**0**	**0**
EDT	**W**	**5**	**7**	**0**	**0**	**0**
ALERTVSM	**E**	**0**	**0**	**0**	**0**	**0**
PAGERVSM	**E**	**0**	**0**	**0**	**0**	**0**
_____	_	**0**	**0**	**0**	**0**	**0**
_____	_	**0**	**0**	**0**	**0**	**0**

Image Filename Field

This field contains the filename of an image that a process would execute to be granted exceptions via their Image Filename. Image Filenames can be explicitly stated and may contain wildcard ("*") values.

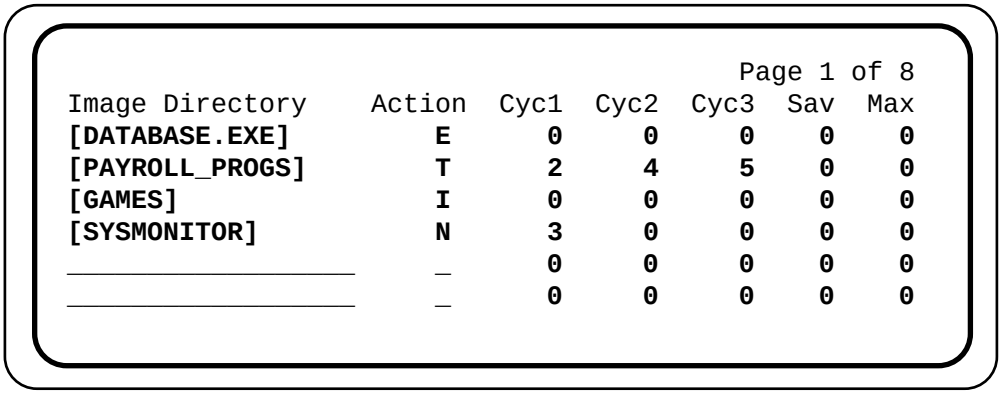
5.19 Sample Image Filename Exception Records

- The first entry shown above makes processes running BACKUP exempt, therefore Assassin will take no action on a process executing this image.
- The second entry shown above makes processes running DISK_MONITOR exempt, therefore Assassin will take no action on a process executing this image.

- The third entry shown above will cause Assassin to send a message to the operator when the process running MAILSENDER is idle for more than 100 minutes (20 cycles x 5 minutes).
- The fourth entry shown above will cause Assassin to immediately terminate any process running STAR_TREK without any warning.
- The fifth entry shown above will cause Assassin to notify the operator when the process running APCHECKPRINT is idle for more than ten minutes (2 cycles x 5 minutes).
- The sixth entry shown above will cause Assassin to warn any process running MAIL after ten minutes (2 cycles x 5 minutes) and twenty minutes (4 cycles x 5 minutes) of idle time and terminate the process after twenty-five minutes (5 cycles x 5 minutes).
- The seventh entry shown above will cause Assassin to warn users who are running EDT and are idle for twenty-five minutes (5 cycles x 5 minutes). Users will also be notified again if they remain idle after thirty-five minutes (7 cycles x 5 minutes).
- The eighth entry shown makes processes running AlertVSM exempt, therefore Assassin will take no action on a process executing this image.
- The last entry shown makes processes running PagerVSM exempt, therefore Assassin will take no action on a process executing this image.

5.20 Image Directory Exception Screen

Figure 5-7 Image Directory Exception Screen



```
                                                    Page 1 of 8
Image Directory     Action  Cyc1  Cyc2  Cyc3  Sav  Max
[DATABASE.EXE]          E     0     0     0    0    0
[PAYROLL_PROGS]         T     2     4     5    0    0
[GAMES]                 I     0     0     0    0    0
[SYSMONITOR]            N     3     0     0    0    0
____________________    _     0     0     0    0    0
____________________    _     0     0     0    0    0
```

Image Directory Field

This field contains the directory specification of an image that a process would execute to be granted exceptions via their image's directory. Directory specifications must be explicitly stated and include all relevant punctuation. Entries **cannot** contain wildcard ("*") values.

5.21 Sample Image Directory Exception Records

- The first entry shown above will cause processes running any program that resides in the directory "[DATABASE.EXE]" to be exempt, therefore Assassin will take no action on these processes.
- The second entry shown above tells Assassin to warn any process running a program that resides in the directory "[PAYROLL_PROGS]" after it has been idle for ten minutes (2 cycles x 5 minutes) and twenty minutes (4 cycles x 5 minutes) and terminate the process after being idle for twenty-five minutes (5 cycles x 5 minutes).
- The third entry shown above will cause Assassin to immediately terminate any process running a program that resides in the directory "[GAMES]".
- The last entry shown above will cause Assassin to notify the operator when any process running a program that resides in the directory "[SYSMONITOR]" is idle for fifteen minutes (3 cycles x 5 minutes).

6 Using ProcessView

6.1 Introduction

Assassin will allow you to continuously view the status of processes on your system. This capability is called ProcessView. ProcessView is an easy-to-use continuous monitor that can show busy processes on your system and includes detailed information about them. You can use ProcessView perform various actions on processes without exiting Assassin.

ProcessView is described in the following sections.

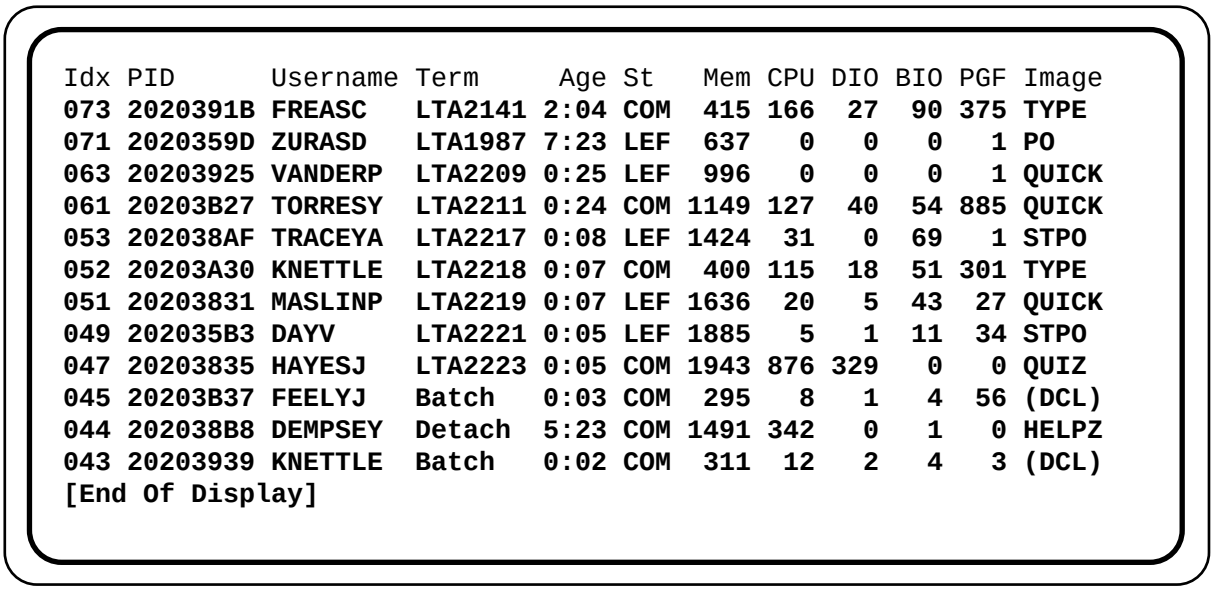
6.2 Starting ProcessView

To start ProcessView, simply issue the following command:

```
$  ASSASSIN /VIEW
```

When you issue this command, Assassin will scan the processes on your system and will produce the display shown below.

Figure 6-1 ProcessView Display Screen



```
Idx PID      Username Term     Age St   Mem CPU DIO BIO PGF Image
073 2020391B FREASC   LTA2141 2:04 COM  415 166  27  90 375 TYPE
071 2020359D ZURASD   LTA1987 7:23 LEF  637   0   0   0   1 PO
063 20203925 VANDERP  LTA2209 0:25 LEF  996   0   0   0   1 QUICK
061 20203B27 TORRESY  LTA2211 0:24 COM 1149 127  40  54 885 QUICK
053 202038AF TRACEYA  LTA2217 0:08 LEF 1424  31   0  69   1 STPO
052 20203A30 KNETTLE  LTA2218 0:07 COM  400 115  18  51 301 TYPE
051 20203831 MASLINP  LTA2219 0:07 LEF 1636  20   5  43  27 QUICK
049 202035B3 DAYV     LTA2221 0:05 LEF 1885   5   1  11  34 STPO
047 20203835 HAYESJ   LTA2223 0:05 COM 1943 876 329   0   0 QUIZ
045 20203B37 FEELYJ   Batch   0:03 COM  295   8   1   4  56 (DCL)
044 202038B8 DEMPSEY  Detach  5:23 COM 1491 342   0   1   0 HELPZ
043 20203939 KNETTLE  Batch   0:02 COM  311  12   2   4   3 (DCL)
[End Of Display]
```

The display contains one line for each process. The information shown is described in the following table.

Table 6-1 ProcessView Display Screen Columns

Column	Description
Idx	A number unique to each process on the system.
PID	The number VMS uses to track each process.
Username	The Username of the process.
ProcessName	The Process Name of the process.
Term	The name of the Terminal for the process.
Age	The age of the process in hours and minutes.
St	The Scheduling State of the Process.
Mem	The total memory pages for the process.
CPU	The CPU Time in hundredths of a second.
DIO	The number of Direct I/O Operations.
BIO	The number of Buffered I/O Operations.
PGF	The number of Page Faults Generated.
Image	The program being executed by the process.
Access Device	Additional information about the Process' device.

6.3 ProcessView Keyboard Keys

The following table lists the keyboard keys active when running ProcessView and the function the keys perform.

Table 6-2 Process View Display Keyboard Capabilities

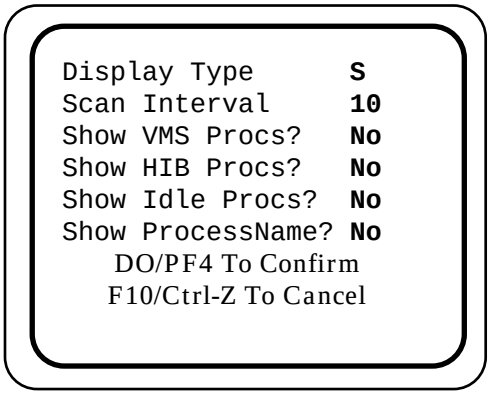
Key Name	Key Function
DO	Cause Assassin to exit from ProcessView
PF4	Cause Assassin to exit from ProcessView
Control-Z	Cause Assassin to exit from ProcessView
Return	Scan Processes on system and display updated information.
Up Arrow	Move scrolling display "up" one line at a time
Down Arrow	Move scrolling display "down" one line at a time
Next Screen	Advance to next display page
Prev Screen	Advance to previous display page
B or b	Display last screen of Scrolling Display
T or t	Display first screen of Scrolling Display
PF2	Provide context-sensitive help about a field or screen
Help	Provide context-sensitive help about a field or screen
F7	Display Assassin's About Screen
F9	Spawn a DCL Subprocess without exiting Assassin
F10	Exit from an entry screen or from Assassin if at the Main Menu
Control-V	Display Broadcast messages in the Message Window
Control-W	Repaint or refresh the terminal screen
Control-N	Advance to next display page

Control-P	Advance to previous display page
A or a	Present ProcessView Actions Form and perform specified action
S or s	Present ProcessView Settings Form to modify ProcessView settings
Any Other	Scan Processes on system and display updated information.

6.4 ProcessView Settings

The behavior of ProcessView's Display Screen is controlled by settings that you can alter. The active settings can be displayed and changed by pressing the "S" key when using ProcessView. The default values for the settings are shown in the following screen. You cannot permanently change the default values because these settings are defined each time ProcessView is started. The fields in the Settings Screen are described below.

Figure 6-2 ProcessView Settings Screen



Display Type

This field defines the type of information displayed by ProcessView. Valid values for this field are "S" to produce the standard display and "D" to produce a display that shows additional information about the process' login device.

Scan Interval

This field defines the screen update frequency. This value is specified in seconds.

Show VMS Procs?

This field defines whether Operating System Processes are displayed. A process is considered to be an Operating System Process if it has a Username of "DECNET" or "AUDIT$SERVER" or a Username of "SYSTEM" and no login device.

Show HIB Procs?

This field defines whether processes in HIB State (Hibernate) are displayed.

Show Idle Procs?

This field defines whether idle processes are displayed.

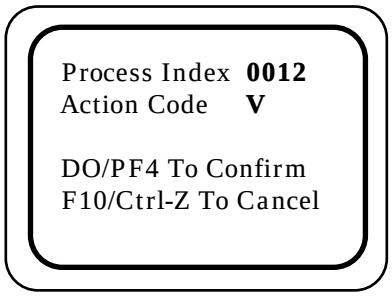
Show Process Name?

This field defines whether the Process Name is shown instead of the Username for each process displayed.

6.5 ProcessView Actions

You can cause ProcessView to perform various actions on processes on your system. To specify an action, press the "A" key when using ProcessView. The Actions Screen will appear and allow you to specify a Process Index and an Action Code. The Process Index for a process is shown in the first column of the ProcessView Display Screen. The fields in the Actions Screen are described below.

Figure 6-3 ProcessView Actions Screen



Process Index

This field allows you to specify a particular process to which the action will be applied. The Process Index for a process is shown in the first Column under the "Idx" heading.

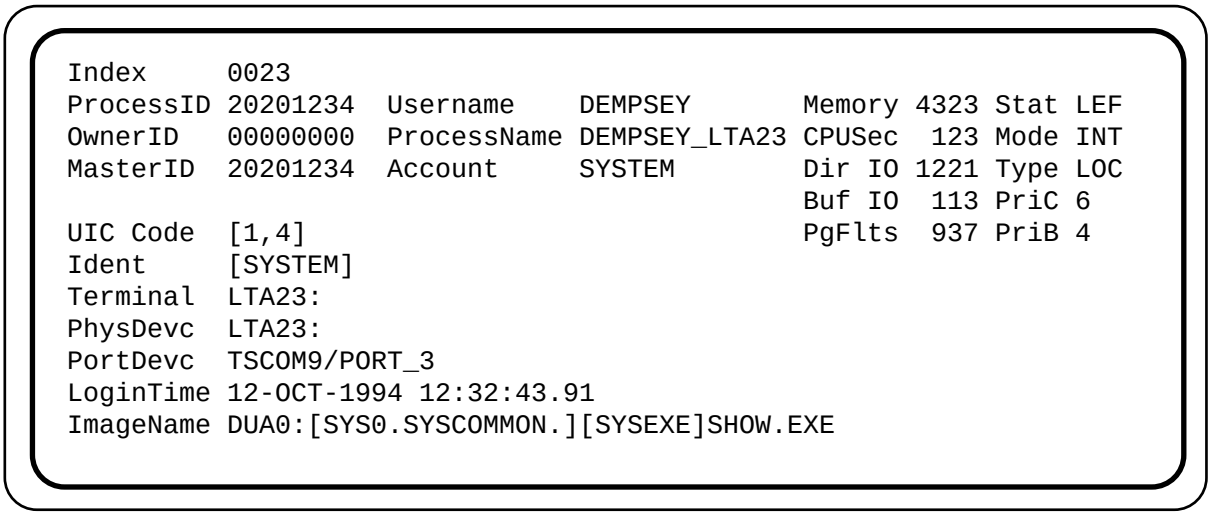
Action Code

This field defines which action ProcessView will perform. Valid actions are "E" to cause an image being executed by the process to exit, "P" to adjust the base priority of a process, "T" to terminate a process, and "V" to view details about a process.

6.6 ProcessView Process Details Screen

The Process Details Screen is shown below.

Figure 6-4 ProcessView Process Details Screen



```
Index     0023
ProcessID 20201234  Username    DEMPSEY       Memory 4323 Stat LEF
OwnerID   00000000  ProcessName DEMPSEY_LTA23 CPUSec  123 Mode INT
MasterID  20201234  Account     SYSTEM        Dir IO 1221 Type LOC
                                              Buf IO  113 PriC 6
UIC Code  [1,4]                               PgFlts  937 PriB 4
Ident     [SYSTEM]
Terminal  LTA23:
PhysDevc  LTA23:
PortDevc  TSCOM9/PORT_3
LoginTime 12-OCT-1994 12:32:43.91
ImageName DUA0:[SYS0.SYSCOMMON.][SYSEXE]SHOW.EXE
```

Table 6-3 ProcessView Process Details Fields

Field	Description
Index	A number unique to each process on the system.
ProcessID	The number VMS uses to track each process.
OwnerID	The Process Id of the process' parent.
MasterID	The Process Id of the process' Master Process.
Username	The Username of the process.
ProcessName	The Process Name of the process.
Account	The value of the UAF Account Field for the process.
Memory	The total memory pages in user by the process.
CPUSec	The CPU Time in hundredths of a second of the process.
Dir IO	The number of Direct I/O Operations of the process.
Buf IO	The number of Buffered I/O Operations of the process.
PgFlts	The number of Page Faults Generated by the process.
Stat	The Scheduling State of the Process.
Mode	The Mode of the process.
Type	The Job Type of the process.
PriC	The Current Priority of the process.
PriB	The Base Priority of the process.
UIC Code	The numeric User Identification Code of the process.
Ident	The alphanumeric Identifier of the process.
Terminal	The name of the Terminal of the process.
PhysDevc	The Physical Device Name of the process.

PortDevc	The Remote Access Port Name of the process.
LoginTime	The date and time that the process was created.
ImageName	The program being executed by the process.

7 Logfiles And Logfile Messages

7.1 Introduction

This chapter describes the two logfiles created by the Assassin Monitor Detached Process.

The first Logfile is the Monitor Logfile and is called ASSASSIN_MONITOR_*nodename*.LOG. It is used to capture messages from the Monitor Processes itself that would normally be written to SYS$OUTPUT or SYS$ERROR. This file is similar to the logfile created by a batch job.

The second Logfile is the Actions Logfile and it is called ASSASSIN_ACTIONS_*nodename*.LOG. It is used to store information about actions taken and notification messages sent by the Monitor Detached Process.

7.2 The Monitor Logfile

This file captures the output or error messages generated by the Monitor Process that would be written to SYS$OUTPUT or SYS$ERROR. This Logfile is named ASSASSIN_MONITOR_*nodename*.LOG. *The node name of your system, if assigned, will be included in the file name.* During normal operation, this file should be empty. If the Monitor Process is not operating correctly, then this file may contain useful diagnostic information. Please review the contents of this file if the Monitor Process does not seem to function correctly. If Assassin continues to operate unsuccessfully or incorrectly, please contact customer support.

This file resides in the same directory as the other Assassin Product Files.

7.3 The Actions Logfile

This file contains messages written by Assassin when Assassin detects idle processes. This file is named ASSASSIN_ACTIONS_*nodename*.LOG. *The node name of your system, if assigned, will be included in the file name.* This Logfile tracks the work Assassin performs and contains valuable information about which processes Assassin is monitoring and what actions have been or will be performed on these processes.

This Logfile can be used to track or identify processes that are being terminated too frequently, or not frequently enough, in order to maximize the use of your system.

Assassin will always add messages to the end of an existing Logfile.

7.4 Creating A New Actions Logfile

To create a new Logfile, simply rename the existing Logfile to a different name (if you want to preserve the contents) or delete the existing Logfile. Assassin will then create the new Logfile automatically when it writes the next message to its Logfile.

7.5 Redirecting The Historical Logfile

This Logfile normally resides in the same directory as the other Assassin Product Files. It can be redirected to another directory by issuing the following command:

```
$ define/system/executive ASSASSIN_ACTIONS_MYNODE  "new-loc"
```

where "new-loc" is the desired device, directory and filename for the new Logfile. The device and directory must already exist and Assassin must have write access to the directory.

Example 7-1 Redirecting Assassin Logfiles To Another Location

Assume that you accepted the default answers to questions asked by the Installation Procedure and installed Assassin on the system disk. This means that all Assassin logfiles are created on the system disk. This was acceptable during pre-purchase evaluation because you kept the logfile size manageable by deleting the file each day. Now that you have purchased the software, you would like to keep more logfile history but the system disk is too full for the expected logfile size. Therefore, you want to place the logfile on a different disk. To do this, place the following command in your system startup command file:

```
$define/system/executive ASSASSIN_ACTIONS_ALPHA -
DUA2:[LOGFILES]ASSASSIN_ACTIONS_ALPHA.LOG
```

This example command assumes that your system's nodename is ALPHA and will cause Assassin to create and maintain its logfile information in DUA2:[LOGFILES]ASSASSIN_ACTIONS_ALPHA.LOG. Substitute your actual system's nodename for the reference to ALPHA when you place the command in your system startup command file. You may choose any filename you desire. You are not required to name the file ASSASSIN_ACTIONS_ALPHA.

7.6 Messages And Logfile Content Settings

The following table describes all of the messages written by Assassin and which Logfile Content Settings will record the various individual messages to Assassin's Actions Logfile.

The values shown in the Cont (Logfile Content) Column in the following table indicate which messages are written to the Actions Logfile. If the value of the Logfile Content field in the Operator Notification Screen is "A", then those messages with an "A" in the Cont Column of the table will be written to the Actions Logfile.

The number shown in the Fmt ("Format") Column of the following table correspond to the number Logfile Record Formats described later in this chapter.

Table 7-1 Messages And Logfile Content Settings

Message Label	Cont	Fmt	Meaning
IPM Enter	A	1	Starting Process Processing Phase
IPM Exit	A	1	Ending Process Processing Phase
IPM Idle	A	2	Process Is Idle
IPM Busy Fina	A	2	Process Became Busy
IPM Blind 1	AOF	2	Blindly Terminate Cycle 1
IPM Custom 1	AO	2	Execute Custom Command - Cycle 1
IPM Custom 2	AO	2	Execute Custom Command - Cycle 2
IPM Custom 3	AOF	2	Execute Custom Command - Cycle 3
IPM Exempt 1	AOF	2	Process Is Exempt - Cycle 1
IPM Freeze 1	AO	2	Suspend Process - Cycle 1
IPM Freeze 2	AO	2	Suspend Process - Cycle 2
IPM Freeze 3	AOF	2	Suspend Process - Cycle 3
IPM ImmTrm 1	AOF	2	Immediately Terminate Process - Cycle 1
IPM Lock 1	AO	2	Lock Process - Cycle 1
IPM Lock 2	AO	2	Lock Process - Cycle 2
IPM Lock 3	AOF	2	Lock Process - Cycle 3
IPM Notify 1	AOF	2	Notify Operator - Cycle 1
IPM SilTrm 1	AOF	2	Silently Terminate Process - Cycle 1
IPM SilTrm 2	AO	2	Silently Terminate Process - Cycle 2
IPM SilTrm 3	AO	2	Silently Terminate Process - Cycle 3
IPM Term 1	AO	2	Terminate Process - Cycle 1
IPM Term 2	AO	2	Terminate Process - Cycle 2
IPM Term 3	AOF	2	Terminate Process - Cycle 3
IPM Warn 1	AO	2	Warn Process Cycle 1
IPM Warn 2	AOF	2	Warn Process Cycle 2
MON Create	AOF	3	Monitor Process Created
MON Delete	AOF	3	Monitor Process Deleted
MON Startup	AOF	3	Monitor Process Startup
MON CFGFile	AOF	4	Monitor Configuration File Changed

7.7 Actions Logfile Message Formats

Messages are written to the Actions Logfile (named ASSASSIN_ACTIONS_*mynode*.LOG) in one of the following formats. For the purposes of this manual, the formats have been assigned numbers between one and five. Each message in the table shown above has a corresponding message format. The data written to the Logfile by each format is listed below. The items are listed in the order in which they appear in the Logfile Record (top-to-bottom in this document is left-to-right in the Logfile Record).

Table 7-2 Logfile Record Format 1 Layout

Data Item	Typical Value
Date	23-AUG-1992
Time	21:49:17
Nodename	ALPHA
Monitor Process ID	20200012
Message Label	See Message Table Above
Username	SMITH
Process Name	SMITH_EDT
Terminal Name	LTA1234:

Table 7-3 Logfile Record Format 2 Layout

Data Item	Typical Value
Date	23-AUG-1992
Time	21:49:17
Nodename	ALPHA
Monitor Process ID	20200012
Message Label	See Message Table Above
Idle Count	0003
Action Code	T
Username	SMITH
Process Name	SMITH_EDT
Terminal Name	LTA1234:
Image Name	EDT
Access Port Name	TSCOM3/PORT_2
Process ID	20201234

Table 7-4 Logfile Record Format 3 Layout

Data Item	Typical Value
Date	23-AUG-1992
Time	21:49:17
Nodename	ALPHA
Monitor Process ID	2020005F
Message Label	See Message Table Above
Monitor Username	SMITH

Table 7-5 Logfile Record Format 4 Layout

Data Item	Typical Value
Date	23-AUG-1992
Time	21:49:17
Nodename	ALPHA
Monitor Process ID	2020005F
Message Label	See Message Table Above
Monitor Username	SMITH
Config File Name	Assassin_CFG_ALL_A

8 Messages Sent By Assassin

8.1 Introduction

This chapter described the messages sent to idle processes and the Operator.

8.2 User Messages

When Assassin sends an idle notification message to a user process, it uses information it knows about the specific process and the message text you specify in the Message Text Lines Screen to compose the message.

Assassin includes process specific information in the message so that users with multiple subprocesses can tell which process the message applies to.

An example message is shown below. The values in italics are variable parts of the message and are taken from information Assassin knows about the process.

> Message From Assassin On Node *node* At *date-time*
> About Process *process-id node*::*user* (*process*) On *terminal*
>
> *text line one*
> *text line two*

The values in italics are as follows:

node	The current Nodename
date-time	The current Date and Time
process-id	The Process ID (PID) of the idle process
user	The Username of the idle process
process	The Process Name of the idle process
terminal	The Terminal Device of the idle process
text line one	The first line of message text for the particular cycle
text line two	The second line of message text for the particular cycle

A fully formatted message would look similar to the following:

> Message From Assassin On Node ALPHA At 20-APR-1992 17:33:37.12
> About Process 2020005F ALPHA::SMITH (SMITH_EDT) On LTA1234:
>
> Please use the VAX or log off the system. When you are idle
> you still consume resources and slow down the VAX system.

8.3 Operator Notification Messages

Assassin sends Operator Notification Messages when it takes action on an idle process. The messages are sent in one of three styles each of which is shown below. The Message Style is specified for Idle Processes in the General Parameters screen.

The messages shown below use italics to describe their variable elements. These italicized terms are explained below:

node	The current Nodename
date-time	The current Date and Time
action-mesg	The reason Assassin is sending the message
process-id	The Process ID (PID) of the idle process
idle-count	The number of cycles for which the process has been idle
user	The Username of the idle process
terminal	The Terminal Device of the idle process
process	The Process Name of the idle process
image-name	The Image Name of the program that the process is executing
acc-port	The Port Name (usually the terminal server/port) of the process
login-time	The date and time that the user logged in
virt-device	The Virtual Device Name (VTAnnn)
cfg-file	The Nodename and ID Letter of the active Configuration File

8.3.1 Notification Message Style 1

An example of a Message Style 1 is shown below. The values in italics are explained above.

Message From Assassin On *node* At *date-time*
Idle Process Found - *action-mesg*
process-id idle-cnt user terminal process image acc-port

A fully formatted message would look similar to the following:

Message From Assassin On Alpha At 17-AUG-1991 22:15:58.12
Idle Process Found - Warn Process Send Warning One
2020005F 02 SMITH LTA123: SMITH_EDT EDT TSCOM3/PORT_2

8.3.2 Notification Message Style 2

An example of a Message Style 2 is shown below. The values in italics are explained above.

Message From Assassin On *node* At *date-time*
Idle Process Found - *action-mesg*
Process ID: *process-id*
Idle Count: *idle-cnt*
Username: *user*
Terminal: *terminal*
Process Name: *process*
Login Time: *login-time*

A fully formatted message would look similar to the following:

Message From Assassin On Alpha At 17-AUG-1991 22:15:58.12
Idle Process Found - Warn Process Send Warning One
Process ID: 2020005F
Idle Count: 02
Username: SMITH
Terminal: LTA123:
Process Name: SMITH_EDT
Login Time: 17-AUG-1991 14:12:23.34

8.3.3 Notification Message Style 3

An example of a Message Style 3 is shown below. The values in italics are explained above.

Message From Assassin On *node* At *date-time*
Idle Process Found - *action-mesg*
Process ID: *process-id*
Idle Count: *idle-cnt*
Username: *user*
Terminal: *terminal acc-port virt-device*
Imagename: *image*
Process Name: *process*
Login Time: *login-time*

A fully formatted message would look similar to the following:

Message From Assassin On ALPHA At 17-AUG-1991 22:15:58.12
Idle Process Found - Warn Process Send Warning One
Process ID: 2020005F
Idle Count: 02
Username: SMITH
Terminal: LTA124: TSCOM3/PORT_2 VTA12:
Imagename: EDT
Process Name: SMITH_EDT
Login Time: 17-AUG-1991 14:12:23.34

8.4 Monitor Timestamp Messages

An example of the Monitor Timestamp Message is shown below. The values in italics are explained above.

Assassin Timestamp On *node* At *date-time* Monitor PID: *process-id cfg-file*

A fully formatted message would look similar to the following:

Assassin Timestamp On ALPHA At 17-AUG-1991 22:12 Monitor PID: 2020005F
ALL_A

8.5 Screen Saver Images

The following figures depict Assassin's Screen Saver Images.

The images assume that the current time is 12:23, the current username is SMITH and the current nodename is VAXONE.

Figure 8-1 Screen Saver Image 1

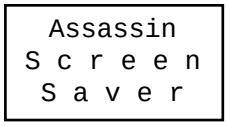


Figure 8-2 Screen Saver Image 2

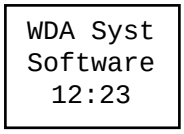


Figure 8-3 Screen Saver Image 3

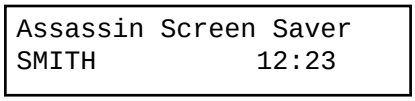


Figure 8-4 Screen Saver Image 4

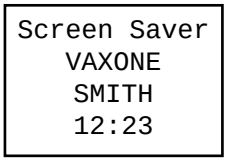


Figure 8-5 Screen Saver Image 5

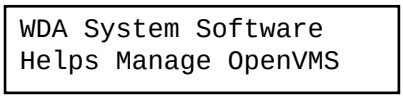
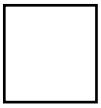


Figure 8-6 Screen Saver Image 6



Appears randomly
in three places on
the screen

DCL Commands

This section describes Assassin's DCL command and its qualifiers. Examples are included at the end of the section.

Format

Assassin

Parameters

The Assassin DCL command does not accept any parameters.

Qualifiers

(no qualifier - Assassin command issued without qualifiers) (Default)

Starts Assassin and displays the Assassin Configuration Menu so that you can create, modify or delete the Assassin's Default Configuration File. Please refer to **Chapters 4, 6, and 7** for more information on how to configure Assassin.

/CHECK_USAGE

Allows users to check their MaxConnect Time Usage and Limit Settings. Please refer to **Section B.8** for more information on how to utilize the MaxConnect Time capabilities provided by Assassin.

/CONFIGURE

Opens the Assassin Configuration Menu so that you can create, modify or delete Assassin Configuration Files for specific nodes, days of the week or times of the day. Please refer to **Chapters 4, 6, 7 and Appendix C** for more information on how to configure Assassin.

/IDENTIFICATION=process-id

Allows you specify the Process ID (PID) of the process to be locked or unlocked.

/INSTALL_KEY=key-string

Allows you to enter a valid Product Key into Assassin. This qualifier is used to extend the expiration date for further evaluation or, after purchasing the product, to enter the official non-expiring Product Key provided by WDA System Software.

/KEYREQUEST

Generates a Product Key Request Form suitable for faxing to WDA System Software. When completed, this form will provide all the necessary product and site information required to generate a permanent non-expiring Product Key. The Product Key Request Form is written to a file called ASSASSIN_KEYREQUEST.LIS in your default directory. You may use the **/OUTPUT** qualifier described below to create the Form in a different location. Fax this form to WDA System Software to

receive a product key.

/LOCK

Allows you to lock your own process manually. This command can be used when you need to leave your terminal unattended and want it to remain secured until you return. Please refer to **Appendix B** for more information.

/MONITOR

This command is normally used only for product support purposes.

Starts the monitoring function of Assassin so that you can view Assassin's operation on your system interactively.

You may choose to view the operation of the Assassin Monitor Process interactively by sending its output to your terminal screen. You can accomplish this by issuing the following command:

```
$ Assassin /MONITOR

Assassin Monitor Process - Logfile Messages Written To Terminal

Performing normal Assassin monitoring function with Logfile
Output directed to your terminal screen. All Configured Actions
and Exceptions will be performed.

Press Control-Y to stop Assassin Monitoring and return to DCL.
```

You should use this technique to evaluate Assassin's operation when you make configuration changes and want to verify the impact of your changes. Press Control-Y at any time to stop viewing Assassin and return to the DCL prompt ("$").

When you test the Monitor Process as described above, a fully functioning Assassin Monitor Process will be executing in the context of your interactive process. **If you configure Assassin to terminate idle processes, then any processes that are idle will be terminated accordingly.**

/NODE=node-name

Allows you to specify the specific node that you want to configure or report. This qualifier is used in conjunction with the **/CONFIGURE** and **/REPORT** qualifiers. If you do not specify these qualifiers, then the value used by Assassin will be "All Nodes".

/OUTPUT=file-spec
/OUTPUT=SYS$OUTPUT: (default)

Defines the name of the output file created when you issue the ASSASSIN/REPORT or ASSASSIN/KEYREQUEST DCL commands. If you omit this qualifier or do not specify the *file-spec* value, then the Configuration Reports are directed to your terminal screen and Product Key Request Forms are created as ASSASSIN_KEYREQUEST.LIS in your default directory.

/REPORT

Causes Assassin to generate a report of the contents of your Assassin Configuration File. This report can be filed for future reference or attached to a Software Performance Report if you request technical assistance.

/RESET

Causes Assassin to reset its configuration data to the Factory Default values.

/START

Starts the Assassin Monitor Detached Process on your system.

The Monitor Process is started as a detached process when you issue the following command:

```
$ Assassin /START

Loading Product Configuration Information...
Loading From Assassin_CFG_DEFAULT_A
Validating Product Configuration Information...

Starting Monitor Detached Process...

Assassin Monitor Process Started Successfully On Node MYNODE
Creation Time   01-MAY-1994 10:31:34.76
Process ID      202005DE
Process Name    Assassin_Monitor
Monitor LogFile DUA0:[ASSASSIN]ASSASSIN_MONITOR_MYNODE.LOG
Actions LogFile DUA0:[ASSASSIN]ASSASSIN_ACTIONS_MYNODE.LOG
```

If the Monitor Process is already running as a detached process on your system, then the existing process will be terminated and a new Monitor Process will be started.

If the existing process takes too long to terminate (about five seconds), then you may receive a "duplicate name" (%SYSTEM-F-DUPLNAM, duplicate name) error message. Simply wait for the existing process to rundown and then reissue the ASSASSIN/START command.

Beware: If the Monitor Process seems to start successfully but doesn't seem to survive, that is, you can't see it on the system, then Assassin may be encountering a problem when it executes the system-wide login command file (SYS$SYLOGIN) or process login command file (LOGIN.COM) for the process that started the Monitor Process. The following DCL command can be inserted AT THE BEGINNING of the relevant file (either in the system-wide or process login command file):

$ if f$mode() .eqs. "OTHER" then $exit

/STOP

Stops the Assassin Monitor Detached Process on your system.

Assassin will provide the following message if you instruct Assassin to stop and it

finds the Monitor Process running on your system:

```
$ Assassin /STOP

Attempting To Stop Monitor Detached Process...
Monitor Detached Process Has Been Stopped
```

Assassin will provide the following message if you instruct Assassin to stop and it **does not** find the Monitor Process running on your system:

```
$ Assassin /STOP

Attempting To Stop Monitor Detached Process...
Monitor Detached Process Was Not Running
```

/TEST

Allows you to test the functionality of the **/CHECK_USAGE** MaxConnect command qualifier. Please refer to **Section B.8** for more information on how to utilize the MaxConnect Time capabilities provided by Assassin.

/TIMEOUT=number-of-minutes

Allows you to specify the number of minutes Assassin's Terminal Lock will wait for you to enter the correct password to unlock your terminal before terminating your process. This can be used with the **/LOCK** qualifier to lock your terminal but to have Assassin terminate your process after some period of time.

/UNLOCK

Allows you to unlock a process that is currently locked by Assassin. You must specify the Process ID of the process to be unlocked by using the **/IDENTIFICATION** qualifier.

/VIEW

Allows you to continuously view information about all processes on your system. Please refer to **Chapter 6** for more information.

A Installation And Deinstallation

A.1 Installation

This section describes how you can quickly and easily install Assassin on your system or in a OpenVMS Cluster environment. A default installation should take about five to ten minutes to complete.

A.1.1 Before You Install Assassin

The installation of Assassin is a three-step process.

1. Determine where you would like Assassin's files to be located. The preferred location is SYS$SYSDEVICE:[ASSASSIN].
2. Use VMSINSTAL to install Assassin onto your system or cluster
3. Edit your system startup command procedures to automatically start the Assassin Monitor Process when your system is rebooted.

A.1.2 Installation Step 1 - Where To Put Assassin's Files

You may place Assassin's Product Files on any disk device that has at least 4,000 free blocks.

You must have sufficient disk quota available on the disk you choose in order to install and operate Assassin successfully.

A.1.3 Installation Step 2 - Using VMSINSTAL

Load the Assassin distribution media onto a suitable device and issue the following command from a privileged account:

```
$ @SYS$UPDATE:VMSINSTAL ASSASSIN device:
```

where "device" is the name of the tape drive that contains the distribution media.

The Installation Procedure will ask for

1. The name of the device and directory to receive Assassin's files. If you use the default, SYS$SYSDEVICE:[ASSASSIN], then the procedure will create the directory. If you use a different device and directory, then the directory must already exist.
2. Permission to insert a help library module into your system-wide Help Library (SYS$HELP:HELPLIB.HLB). This is recommended, although optional, and can be easily accomplished after the installation if you prefer not to insert the Assassin help library module at this time. The default is to insert the Assassin help library module into your system-wide Help Library.
3. Permission to add a DCL command to your system-wide DCL Command Tables (SYS$LIBRARY:DCLTABLES.EXE). This is recommended, although optional,

and can be easily accomplished after the installation if you prefer not to insert the Assassin DCL command at this time. The default is to add the Assassin DCL command to your system-wide DCL Command Tables.

A.1.4 Installation Step 3 - Start The Monitor Process At System Startup

Place the following command in your system startup procedure so that Assassin's Monitor Process will be started automatically when your system is rebooted. The startup file is usually named SYS$MANAGER:SYSTARTUP_VMS.COM.

```
$ ASSASSIN /START
```

A.1.5 How To Install Assassin's DCL Command After Installation

To install Assassin's DCL Command into your systemwide DCL Command Tables after the initial installation of Assassin, please perform the following steps.

1. Ensure that all users are off the system.
2. Execute the command file called ASSASSIN_INSTALL.COM that can be found in the Assassin Product Directory. Enter "Yes" when asked if you want to install Assassin's DCL Command.

A.1.6 How To Install Assassin's Help Library Module After Installation

To install Assassin's Help Library Module into your systemwide Help Library after the initial installation of Assassin, please perform the following steps.

1. Ensure that all users are off the system.
2. Execute the command file called ASSASSIN_INSTALL.COM that can be found in the Assassin Product Directory. Enter "Yes" when asked if you want to install Assassin's Help Library Module.

A.2 Installing Assassin In A Cluster Environment

To install Assassin in a cluster environment, simply perform the following steps:

1. Install Assassin as described above from one of the nodes in the cluster. Be sure to choose a Product File disk and directory that is accessible to all nodes in the cluster.
2. For each of the other nodes on which you intend to use Assassin, execute the command file named ASSASSIN_INSTALL.COM that can be found in the Assassin Product Directory. Be sure to answer "Yes" when asked about installing Assassin's DCL Command and Assassin's Help Library Module onto each system.

A.3 Deinstallation

This section describes how you can easily and completely remove the Assassin files from your system.

A.3.1 Before You Deinstall Assassin

The only thing you need to know to remove Assassin is the location of Assassin's Product Files. If you accepted the default location during installation, then the location would be SYS$SYSDEVICE:[ASSASSIN].

If you desire, you may save Assassin's Configuration Files prior to executing this command. To do so, simply copy the Configuration Files to another location on your system. The Configuration Files to copy are are called ASSASSIN_CFG_*.DAT

A.3.2 How To Deinstall Assassin

To remove Assassin and all of the Assassin Product Files from your system or from its current location in order to install it in another location, please issue the following command:

```
$ @ SYS$SYSDEVICE:[ASSASSIN]ASSASSIN_DEINSTALL
```

If you installed Assassin in a location other than "SYS$SYSDEVICE:[ASSASSIN]", please enter the location you originally chose instead. This command procedure will verify that the Assassin program resides in the directory you specify and then continue as shown in the following section. Once the deinstallation procedure has been completed, then Assassin has been completely removed from your system.

B Miscellaneous Topics

B.1 Producing A Configuration File Report

You can easily print a report showing the contents of Assassin's Configuration Files. To generate a report simply issue the following command:

```
$ Assassin /REPORT [/NODE=nodename] [/OUTPUT=filespec]
```

You may include the optional /NODE or /OUTPUT qualifiers to show information pertaining to other nodes or to write the report to a desired filename.

Please consider generating a report of the configuration files and filing it for future reference. You will then be able to compare configurations over time as the needs of your site change.

Please include a copy of the Configuration File Report when reporting Software Problems to WDA System Software or your software distributor.

B.2 Moving The Product Files

You can move Assassin's Product Files to another location on your system, if necessary, by performing the following:

1. Stop the Assassin Monitor Process by issuing the DCL command Assassin/STOP.
2. Reinstall Assassin using the original media in the desired location. **Be sure to install the DCL command in your system's DCLTABLES when prompted by the Installation Procedure.**
3. Copy the Configuration Files from the old location to the new location. The Configuration Files are named ASSASSIN_CFG_*node_letter*.DAT, where *node* and *letter* refer to the particular files you have created. The default file name is ASSASSIN_CFG_ALL_A.DAT. This step will preserve all of your customizations allowing you to continue operating as before, without requiring re-entry of the various exceptions and changes you may have previously entered.
4. Restart the Assassin Monitor Process by issuing the DCL command ASSASSIN/START.

B.3 Assassin's Keyboard Capabilities

The following keys on an LK201 (VT200-style) keyboard provide the functionality shown in the following table.

Table B-1 Assassin's Keyboard Capabilities

Key Name	Key Function
DO	Cause Assassin to exit from an entry screen saving entries
PF4	Cause Assassin to exit from an entry screen saving entries
Return	Validate and store entry into a field and move to the next field
Up Arrow	Move cursor "up" one field
Down Arrow	Move cursor "down" one field
Right Arrow	Move cursor inside field to the right or to next field
Left Arrow	Move cursor inside field to the left or to previous field
Help	Provide context-sensitive help about a field or screen
F7	Display Assassin's About Screen
F9	Spawn a DCL Subprocess without exiting Assassin
F10	Exit from an entry screen or from Assassin if at the Main Menu
Control-V	Display Broadcast messages in the Message Window
Control-W	Repaint or refresh the terminal screen
Control-Z	Cause Assassin to exit from an entry screen saving entries
Control-N	Advance to next screen page
Control-P	Advance to previous screen page
Next Screen	Advance to next screen page
Prev Screen	Advance to previous screen page

B.4 How Assassin Terminates Processes

The following operations take place when Assassin terminates an idle process:

1. Assassin broadcasts a clear-screen message to the user's terminal screen to ensure that no sensitive information remains after the process is terminated. This step is not performed if the Action Code is "B" (Blindly Terminate).
2. Assassin waits five seconds for the clear-screen message to arrive and take effect.
3. Assassin broadcasts the third action message to the user's terminal screen. This step is not performed if the Action Code is "B" (Blindly Terminate), "I" (Immediately Terminate) or "S" (Silently Terminate).
4. Assassin again waits five seconds for the message to arrive and the terminal to display it.
5. Assassin issues a command to the idle process that causes the process to exit the image it is running so that any exit functions the image would normally perform are completed. This allows the cleanest possible exit from the image being run.
6. Assassin waits five seconds for the image to exit as cleanly as possible.
7. Assassin issues a command to the idle process that causes the process to delete itself. It is important to note that if the process is unable to respond to requests from the system, then it may not delete itself immediately. This limitation is associated with the VMS operating system and not with Assassin.

B.5 Determining Idle Threshold Values

The General Parameters Screen allows you to determine what level of activity is considered idle at your site for your applications. The default values are set so the that processes must be completely idle before Assassin will take any action on them.

B.5 Determining Idle Threshold Values

You may notice that certain process are not deleted when you would like them to be. This could be caused by the process consuming small amounts of resources that keep it from being considered idle but are too small to be noticed. Certain programs may send a message to the terminal on a timed basis or otherwise operate when the user is "idle", therefore making the process active when there is no actual user activity.

To adjust for this situation, simply set the thresholds higher so that the low level of resource consumption by the process being considered falls below the idle thresholds listed in the General Parameters Screen.

You can determine the level of resource consumption for a process by issuing the following DCL command:

```
$ SHOW PROCESS /ACCOUNTING /ID=pid
```

where "pid" is the Process ID of the process you are interested in examining. Note the accounting values for CPU Time, Direct and Buffered I/O and Pagefault Counts. After a sufficient period of time has elapsed (preferably the time specified as your cycle length), re-issue the command and compare the values. The difference in the value of the resources represents the process resource consumption. This usage amount will be useful when setting the new threshold values. You will want your specified threshold value to be greater than the usage amount if you want the process terminated. Monitoring the resource consumption over several cycles will yield the most accurate resource consumption information.

Example B-1 Determining Process Resource Usage

You have a process that seems to be idle and you believe should be terminated by Assassin, however, it is not terminated. The Process ID for the process in question is 2020005F. To determine if the process is indeed idle, issue the following command:

```
$  SHOW PROCESS /ACCOUNTING /ID=2020005F

23-AUG-1992 12:40:41.43  User: SMITH   Process ID:  2020005F
                         Node: ALPHA   Process name: "SMITH"

Accounting information:

Buffered I/O count:  154499  Peak working set size:  8200
Direct I/O count:    121361  Peak virtual size:     14979
Page faults:         509261  Mounted volumes:          11
Images activated:      1014
Elapsed CPU time:    0 00:46:44.68
Connect time:        0 02:55:44.06
```

Wait five minutes (or the value of your Cycle Length Setting) and re-issue the command as shown:

```
$  SHOW PROCESS /ACCOUNTING /ID=2020005F

23-AUG-1992 12:45:41.45  User: SMITH   Process ID:  2020005F
                         Node: ALPHA   Process name: "SMITH"
```

```
Accounting information:

Buffered I/O count:  154499  Peak working set size:  8200
Direct I/O count:    121361  Peak virtual size:     14979
Page faults:         509261  Mounted volumes:          11
Images activated:      1014
Elapsed CPU time:    0 00:46:44.88
Connect time:        0 03:00:44.08
```

This example shows that the process has consumed two-tenths of a second of CPU Time (0 00:46.44.88 - 0 00:46:44.68 = 0 00:00:00.20). The process has not generated any I/O or Pagefault activity.

Therefore, for this example, you should consider increasing the CPU Time Threshold in the General Parameters Screen by 21. Then Assassin will "mark" that the process is idle and proceed normally.

B.6 Send Mail Messages Without Using VMSmail

The DCL command constructed to send Mail Messages is shown below.

```
MAIL /subject="mail-subj" message-file username
```

"MAIL"	is the command entered in this field
"mail-subj"	is the subject created by Assassin
"message-file"	is a file created by Assassin containing the message
"username"	is the address to which the message is to be sent

You may need a DCL command file that parses this command line to provide appropriate data to your Mail Sending Agent. Simply create a command file accessible to Assassin and enter an at-sign ("@") followed by the name of the command file in the mail address field. Include the device and directory, explicitly or via logical name, so Assassin will know where to find the file. If you created a file named "MY_SENDER.COM" located in "DUA0:[JONES]", then you would enter "@DUA0:[JONES]MY_SENDER.COM" in this field.

Note: If the Mail DCL Command is "MAIL", then Assassin will use the callable mail capability of VMS. In all other cases, Assassin will spawn the mail command you specify. If the mail command is spawned, then the process of mailing messages on your system could be slower, depending on the current load on your system and the efficiency of the Mail Sending Agent.

B.7 Determining Your Cycle Length

You may set the Cycle Length to any value you desire that adequately meets your needs. Assassin uses a default value of five ("5") minutes.

You can reduce the resources consumed by Assassin by choosing the longest Cycle Length suitable to the Actions you want Assassin to perform.

Assume that you want to warn your users once after they have been idle for thirty ("30") minutes, again after they have been idle for sixty ("60") minutes and then terminate them after they have been idle for ninety ("90") minutes. The following

table shows various Cycle Lengths and Cycle Counts that you could use to tell Assassin to behave as desired.

Table B-2 Sample Cycle Lengths And Associated Cycle Counts

Cycle Length	First Action	Second Action	Third Action
1 minute	30	60	90
2 minutes	15	30	45
5 minutes	6	12	18
10 minutes	3	6	9
15 minutes	2	4	6
30 minutes	1	2	3

The most optimal values to use when configuring Assassin are those shown on the last line of the table. Using these entries, Assassin will consume the lowest CPU and Logfile disk space resources on your system and still accomplish your goals, given the scenario described above. However, as the cycle length is increased, the variable time before a process is identified as idle will be increased.

B.8 How MaxConnect Time Processing Works

Assassin can terminate processes that have been logged for an excessive period of time in a given calendar day. The calendar day starts and ends at midnight. Some sites may utilize this capability to prevent users from occupying terminals in a high demand area or environment, such as universities.

Assassin will allow you to configure the number of minutes a process can be logged in during any one calendar day using both a default value as well as exception record values. The desired daily time limit is expressed as a number of Monitoring Cycles. You may set the default value for the MaxConnect Time Cycle Count to be one value and have different values for different exception records.

When a process is nearing its MaxConnect Time Limit, Assassin will send at least two messages to the process warning it that it will be terminated soon. Assassin will send a different, final message just before the process is terminated. You may provide your own message text in the Message Text Lines Screen. The warning messages are sent at two cycles and at one cycle before termination.

The Monitor Detached Process tracks Connect Time for all users with MaxConnect Time Limits. If the Monitor Detached Process is stopped and then restarted, then the usage statistics for your users will be lost.

Any process that has a MaxConnect Time Limit value will be terminated whether it is idle or not. Please exercise caution when implementing this capability. If a user's process is terminated before they have saved any pending work, then that work can be lost depending on the capabilities of your application.

Example B-2 MaxConnect Time Limit Example

Assume that your Monitoring Cycle Length is five minutes and you need to limit processes to a maximum of two hours of computer usage

1. Set the MaxConnect Time Cycle Count field to "24" (2 hours divided by 5 minutes).
2. Assassin would then warn processes when they have been logged in longer than 22 cycles.
3. Assassin would terminate processes that have been logged in longer than 24 cycles.

If you change the MaxConnect Time Limit value at a time when users are active on the system, then the affected users will be terminated after they have been logged in longer than the new value . Assassin continually compares a processes limit with its connect time to determine what action to perform.

B.8.1 Check User Connect Time At Login

The following command fill check a user's Connect Time Usage and Limit and optionally terminate the user's process if their Connect Time Usage exceeds their Limit.

```
$  Assassin /CHECK_USAGE [/TEST]
```

If the /TEST qualifier is present, then the user's process will not be terminated.

Example B-3 User Connect Time Check At Login

```
$ Assassin /CHECK_USAGE

Assassin CONNECT TIME STATUS REPORT
For User SMITH As Of 01-MAY-1994 12:34
Connect Time Limit 120 Minutes
Connect Time Usage 125 Minutes

Assassin Connect Time Limit Exceeded
Process Will Be Terminated Now
```

Note: Users that execute this command must be able to read the file ASSAS-SIN_USAG_nodename. DAT that is created in the directory that contains Assassin's files. Usually providing Execute access to the World user group is sufficient.

B.9 How Assassin Performs Screen Saver Functions

Assassin can perform Screen Saver processing on terminals of idle processes. Some security conscious sites may utilize this capability to prevent others from viewing the contents of unattended terminal screens. When a process is idle, Assassin will clear the process's terminal screen and display the desired Screen Saver Image on the terminal.

Assassin will allow you to configure the Image Number using both a default value as well as exception record values. The Images available to you are shown in **Chapter 10**.

Assassin performs its Screen Saver processing only at the end of each monitoring cycle. If your Monitoring Cycle Length is five minutes, then users will notice their screens changing no more often than every five minutes.

Assassin will not perform Screen Saver Processing on the following types of processes:

1. Processes that are not using terminals.
2. Processes that have subprocesses.
3. Processes that are subprocesses.
4. Processes that are locked.
5. Processes that are exempt.

WDA System Software is very interested in its customer's comments or suggestions about this feature.

B.10 AssassinLock Terminal Lock Processing

Assassin can lock the terminals of idle, unattended processes. This feature is implemented as an Action Code "L" - Lock Terminal. Please refer to **Chapter 3** for more information about Actions performed by Assassin. You must define the default or exception Action Code to "L" and define Cycle Counts appropriate to your needs.

- The First Action Cycle Count defines when Assassin will send the First Action message. Assassin will send the message when the process has been idle for the number of cycles specified by the First Action Cycle Count. Set this field to zero ("0") if you do not want this message to be sent.
- The Second Action Cycle Count defines when Assassin will lock the idle process's terminal. Assassin will send the Second Action message and then lock the terminal when the process has been idle for the number of cycles specified by the Second Action Cycle Count.
- The Third Action Cycle Count defines the amount of time that the user is allowed to provide the unlock password. If the user does not unlock the terminal in this amount of time, then Assassin will terminate the process. Set this field to zero ("0") if the user has an unlimited amount of time to unlock the terminal.

When Assassin locks the user's terminal, any processes attached to that terminal will be stalled and will perform no work until the terminal is unlocked.

If the user's User Authorization File (SYSUAF) record is configured with two passwords, then both passwords must be specified to unlock the terminal.

Assassin's terminal locking capability will create one subprocess for each locked terminal. The subprocess is identified as "SnpLck_*pid*" where *pid* is the Process ID of the locked process. You may have to increase the PRCLM value in the User Authorization File (SYSUAF) of the account(s) that start the Assassin Monitor Detached Process if there are/will be a large number of processes locked at any one time.

B.10.1 Unlocking A Locked Terminal

The following command will unlock a terminal that has been locked by Assassin. The command assumes that the Process ID (PID) of the locked process is 20201243.

```
$  Assassin /UNLOCK /IDENTIFICATION=20201243
```

Example B-4 Unlocking A Locked Terminal

```
$ Assassin /UNLOCK /IDENTIFICATION=20201243

AssassinLock - Unlocking Process
Process ID:     20201243
Username:       SMITH
Process Name:   SMITH_A1
Terminal:       LTA1234:

Okay To Unlock This Process? [No]: YES

AssassinLock - Process Has Been Unlocked
```

C Assassin Configuration Files

C.1 Introduction

Assassin allows you to create multiple Configuration Files to better meet the needs of your organization. You can create specific Configuration Files for individual nodes in your OpenVMScluster, for specific times of the day or days of the week. Creating multiple Configuration Files is completely optional since Assassin allows you to maintain a Default Configuration File.

Assassin's Configuration utilities can be accessed by either of the following commands:

```
$  Assassin
```

or

```
$  Assassin /CONFIGURE
```

The presence of the /CONFIGURE Command Qualifier determines the Assassin Configuration File selected. If the qualifier is absent, Assassin will allow you to work only with the Default Configuration File. If the qualifier is present, Assassin will allow you to work with multiple Configuration Files.

Unless you desire Assassin to function differently on selected nodes, at specific times or on specific days, you will find it easier to use the Default Configuration File capabilities.

It is recommended that you not make Assassin's Configuration more complicated than is required to meet the needs of your organization.

C.2 Format Of Configuration File Filenames

When you save or store configuration information, Assassin creates a Configuration File Name in the following format:

ASSASSIN_CFG_*nodename_letter*.DAT

where *nodename* represents the node to which the information applies and *letter* is a single character between "A" and "O" that simply defines the unique Configuration File. A *nodename* value of "ALL" applies to all nodes and is the default value used by Assassin. The default value for *letter* is "A".

The name of the Default Configuration File is ASSASSIN_CFG_ALL_A.DAT.

C.3 How the Monitor Process Finds Configuration Files

When the Assassin Monitor Process looks for configuration data, it first tries to find a Configuration File that is valid for the current node, day and time. If the Monitor Process cannot find a node-specific file, then it tries to find a Configuration File that is valid for the current day and time for ALL nodes. If it cannot find a Configuration

File that is valid for the current day and time, then Assassin will "hibernate" until a Configuration file is valid.

C.4 Managing Configuration Files

The following sections describe how to perform various Configuration File functions.

C.4.1 Maintaining The Default Configuration File

Assassin's Default Configuration File is maintained by using the Assassin DCL command without the /CONFIGURE Command Qualifier. Any entry or change entered into the configuration screens when Assassin is started without using the /CONFIGURE qualifier will be written to the Default Configuration File when you exit the Configuration Menu.

C.4.2 Creating New Configuration Files

A new Configuration File is created by changing the Output File Nodename and/or Output File ID Letter on the Configuration File Setup Screen. When creating a new Configuration File, you may specify a different set of days or hours during which the Configuration File should be used by Assassin. Enter a specific Nodename to create a node specific Configuration File for that individual node. Enter a different ID Letter to create a Configuration File that is valid during a different period of time.

Example C-1 Creating A Node Specific Configuration Files

Suppose that you need to create, for the first time, a Configuration File specific to your production node which is named BIGVAX. If you enter the command:

```
$ Assassin /CONFIGURE /NODE=BIGVAX
```

then Assassin's configuration screens will be started so you can enter your configuration information. When you exit the Configuration Main Menu by selecting the "Z" ("Exit and Save Changes") choice Assassin will save your configuration in a file named Assassin_CFG_BIGVAX_A.DAT. When the Assassin Monitor Detached Process is started on node BIGVAX, it will read the information in this file rather than the Default Configuration File.

C.4.3 Modifying Existing Configuration Files

To modify an existing Configuration File, simply perform the following steps:

1. If the Configuration File is for the Default Node (all nodes), simply issue the command $ASSASSIN/CONFIGURE.
2. If the Configuration File is for a named node, simply issue the command $ASSASSIN/CONFIGURE/NODE=*mynode* where *mynode* denotes the particular node whose configuration you wish to change.
3. Select the appropriate Input File ID Letter from the Configuration File Selection Screen.
4. Update the Configuration File Setup Screen as desired and press the DO or PF4 key and Assassin will display the Main Configuration Menu.

C.4.4 Deleting Configuration Files

To delete an existing Configuration File, simply perform the following steps:

1. If the Configuration File is for the Default Node (all nodes), simply issue the command $ASSASSIN/CONFIGURE.
2. If the Configuration File is for a named node, simply issue the command $ASSASSIN/CONFIGURE/NODE=*mynode* where *mynode* de
3. s the particular node whose configuration you wish to change.
4. Then, select the appropriate Input File ID Letter from the Configuration File Selection Screen.
5. Enter a "Y" in the "Delete Input File?" field and press the DO or PF4 key. The selected Configuration File will be deleted and Assassin will present the Configuration File Selection Screen for another selection.

C.5 Configuration File Screens

Figure C-1 Configuration File Menu Screen

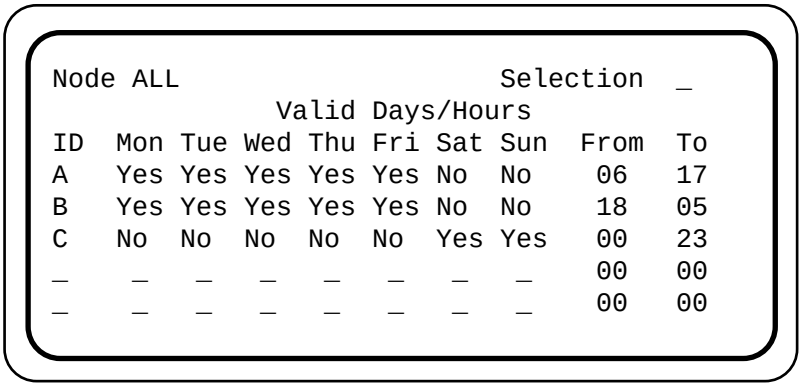
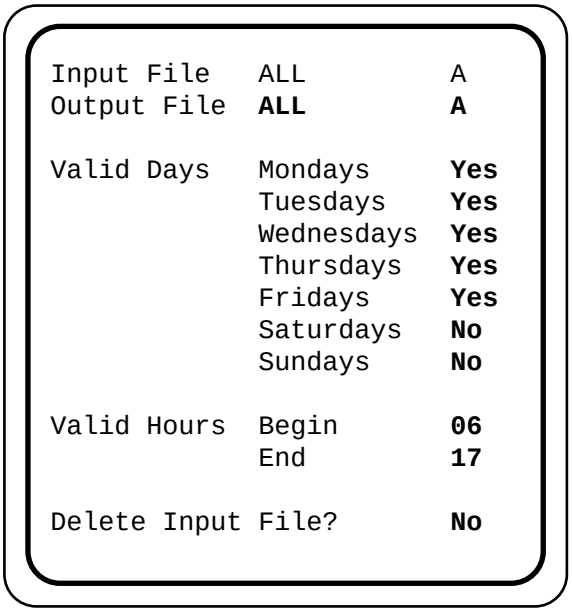


Figure C-2 Configuration File Setup Screen



C.5.1 Data Fields

Selection

This field allows you to specify the ID of the Configuration Input File that you would like to modify or use as a seed (or source) for a new Configuration File. The value entered in this field must be one of the values listed in the ID Field which is described next.

ID

This field lists the ID Letter of the various Configuration Files that have been created for the node selected. The node selected is always shown in the upper-right corner of the screen. You must enter one of the ID Letters shown in the Selection Field.

`Default:` "A" - The Default Configuration File ID Letter is "A".

Output File Nodename

This field accepts the Nodename Assassin will use when saving or storing the Configuration File. This field allows you to save configuration information in a Configuration File for individual nodes.

`Default:` Assassin will use the value of the Input File Nodename.

Output File Letter

This field accepts the ID Letter Assassin should use when saving or storing the Configuration File. This field allows you to save configuration information in an individual filename.

Default: Assassin will use the value of the Input File ID Letter.

Valid Days

This field describes on which days a particular Configuration File is valid and therefore will be used by Assassin. Enter a "Y" if the Configuration File is valid on the particular day or an "N" if it is not valid. For example, to indicate that Tuesday is valid, enter a "Y" in the Tuesday Valid day field. To indicate that weekdays are valid enter a "Y" in each of the Monday through Friday fields.

Default: All "Yes" - The Default Configuration File is valid everyday.

Valid Hours Begin/End

These fields define the range of hours during which a particular configuration file is valid and should be used by Assassin. Please enter the hour a Configuration File should become valid in the Begin Field and the hour in which a Configuration File should become invalid in the End Field. To specify a valid hour range of 8:00 am to 5:00 p.m., enter "08" in the Begin Field and "17" in the End Field. To specify a valid hour range of 8:00 p.m. to 6:00 am enter "20" in the Begin Field and "06" in the End Field.

Default: "00" and "23" - The Default Configuration File is valid all day.

Delete Input File?

This field allows you to delete the Input Configuration File. Enter "Y" and press the DO or PF4 key to delete the Input Configuration File or "N" to not delete the file. If you enter "Y", Assassin will delete the Input Configuration File and present the Configuration File Selection Screen for your selection of the next Configuration File to be modified.

Default: "No" - Assassin will not delete the Input File.

C.5.2 Defining Valid Hours

The following table shows what you should enter for selected ranges of hours during which a Configuration File should be valid and used by Assassin.

Table C-1 Valid Hours Values

Desired Range Of Times	Begin	End
8:00 a.m. to 6:00 p.m.	08	17
6:00 p.m. to 8:00 a.m.	18	07
6:00 p.m. to 12:00 midnight	18	23
12:00 midnight to 12:00 noon	00	11
10:00 p.m. to 7:00 a.m.	22	06
1:00 a.m. to 9:00 a.m.	01	08

C.6 Sample Configurations

The following screens show sample Configuration File entries and the days and times that each configuration is valid.

1. Assassin /CONFIGURE

```
Node ALL                              Selection  _
                    Valid Days/Hours
ID  Mon Tue Wed Thu Fri Sat Sun  From  To
A    Y   Y   Y   Y   Y   N   N   08   17
B    Y   Y   Y   Y   Y   N   N   18   23
C    Y   Y   Y   Y   Y   N   N   00   07
D    N   N   N   N   N   Y   Y   00   23
_    _   _   _   _   _   _   _   00   00
_    _   _   _   _   _   _   _   00   00
```

File A is valid from 8:00a. m. to 6:00p.m. Monday through Friday.
File B is valid from 6:00p.m. to 12:00p.m. Monday through Friday.
File C is valid from 12:00p.m. to 8:00a.m. Monday through Friday.
File D is valid all day on Saturday and Sunday.

2. Assassin /CONFIGURE /NODE=ALPHA

```
Node ALPHA                          Selection  _
                  Valid Days/Hours
ID  Mon Tue Wed Thu Fri Sat Sun  From  To
A    Y   Y   Y   Y   Y   N   N    08   17
B    Y   Y   Y   Y   Y   N   N    18   07
C    N   N   N   N   N   Y   N    00   23
D    N   N   N   N   N   N   Y    00   23
_    _   _   _   _   _   _   _    00   00
_    _   _   _   _   _   _   _    00   00
```

File A is valid from 8:00a.m. to 6:00p.m. Monday through Friday.
File B is valid from 6:00p.m. to 8:00a.m. Monday through Friday.
File C is valid all day on Saturday.
File D is valid all day on Sunday.

The entries shown in example one are valid for all nodes while the entries shown in example two are valid for Node ALPHA only.

Glossary

Account

The value of the Account field in the UAF record for a user. This value can be used to create Exception Records using the Account Exception Record screen.

Action

What we call the work that Assassin performs when it encounters an idle process. Valid Actions include "Terminate", "Notify Operator" or "Warn User". Please refer to **Chapter 3** for more information about the supported Actions provided by Assassin.

Action Code

The letter used to represent the action Assassin is to perform. See **Chapter 3** for a list of valid Action Codes.

Broadcast Message

A message sent by Assassin using the $BRKTHRU VMS System Service.

Cycle Count - First Action

The number of cycles for which a process must be idle before the First Action is performed on it.

Cycle Count - Second Action

The number of cycles for which a process must be idle before the Second Action is performed on it.

Cycle Count - Third Action

The number of cycles for which a process must be idle before the Third Action is performed on it.

Cycle Length

The number of minutes between the sampling passes performed by Assassin. Assassin will gather data about your system periodically where the time between samplings is defined by the Cycle Length.

Default Action Code

The Action Code Assassin will apply to a process unless it determines that the process is covered by an exception record.

Detached Process

A process that is not attached to a terminal.

Dialup Process

A process that accesses the system via a dialup terminal device. OpenVMS determines the type of the process when the process first logs in.

Exception Record

A record that tells Assassin not to perform the Default Action on an idle process. The Exception Record tells Assassin what action to perform and when.

Exempt Process

A process that will never be affected by Assassin no matter how long it is idle.

Glossary

First Action

The sub-action Assassin will perform when Assassin encounters a process that has been idle for the number of cycles specified by the First Action Cycle Count. See **Chapter 3** for more information.

Idle

A condition where the process consumes no CPU Time and generates no I/Os or Pagefaults.

Idle Threshold

An amount of CPU Time or number of I/Os or Pagefaults that defines how idle a process must be before Assassin will determine it to be idle.

Image Filename

The filename part of the image (program) the process is executing. For example, if a process is running a program called "DUA0:[UTILS.EXE]VALIDATE.EXE", then the Image Filename is "VALIDATE".

Image Directory

The directory part of the image (program) the process is executing. For example, if a process is running a program called "DUA0:[UTILS.EXE]VALIDATE.EXE", then the Image Directory is "[UTILS.EXE]".

Local Process

A process that accesses the system via a local terminal device. OpenVMS determines the type of the process when the process first logs in.

Mail Sending Agent

Software that performs mail sending and receiving functions. Examples include VMSmail, All-In-1, and various third-party products.

Message One

The text of a message sent to the idle process by Assassin when the First Action is performed on it.

Message Style

A value that determines how much information Assassin should send to the Operator when it performs an Action. Values range from one ("1") to three ("3") where one provides brief information and three provides the most information. A value of zero ("0") tells Assassin not to send a message at all.

Message Text

The text of a message sent to the idle process (user's terminal) by Assassin.

Message Three

The text of a message sent to the idle process by Assassin when the Third Action is performed on it.

Message Two

The text of a message sent to the idle process by Assassin when the Second Action is performed on it.

OPCOM Message

A message sent by Assassin to the Operator Communication (OPCOM) process. OPCOM can then forward the message depending on your configuration.

Operator

The System Operator or System Manager who normally controls and monitors the configuration and operation of Assassin.

Operator Console

This is any terminal where the user has issued the DCL command "REPLY/ENABLE". This is also the OPA0: System Console Device.

Process Name

The name of the process. This value can be viewed by issuing the DCL command "SHOW USERS/FULL".

Remote Process

A process that accesses the system via DECnet using the "SET HOST" DCL command from another network node. OpenVMS determines the type of the process when the process first logs in.

Second Action

The sub-action Assassin will perform when Assassin encounters a process that has been idle for the number of cycles specified by the Second Action Cycle Count. See **Chapter 3** for more information.

Third Action

The sub-action Assassin will perform when Assassin encounters a process that has been idle for the number of cycles specified by the Third Action Cycle Count. See **Chapter 3** for more information.

UIC

The standard abbreviation for a User Identification Code.

VMSinstal

A command file included in the operating system used to install software on OpenVMS systems. The command file is located in the SYS$UPDATE directory.

Wildcard Character

A character used to create a pattern which is used to match entries. The only wildcard value supported by Assassin is the asterisk ("*").

Index