

ConsoleAlert

NetWare Fileserver Console Reporting System.

Version 1.0

- DEMONSTRATION VERSION ONLY -

For NetWare Versions 4.x and 5.x

Manual Revision : June, 1998

**Protocom Development Systems
Innovation
and
Security**

Disclaimer

Protocom Development Systems makes no representations or warranties with respect to the contents of this manual, and specifically disclaims any express or implied warranties or fitness for any particular purpose. Further, Protocom Development Systems reserves the right to revise this publication and to make changes to its contents, at any time, without obligation to notify any person or entity of such revisions or changes.

© Copyright 1993, 94, 95, 96, 97, 98 Protocom Development Systems Ltd.
All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express prior written consent of the publisher.

Protocom Development Systems Limited.
Suite 2, 240 North Wellwood Avenue
Lindenhurst NY 11757
United States.

Protocom Development Systems Limited.
Old Mill Barn, Mill Lane
Frampton Cotterell
Bristol BS17 2AA
United Kingdom.

Protocom Development Systems Pty Limited.
P.O Box 1101.
Tuggeranong Mail Centre, ACT 2901.
Australia.

Australia	Telephone +61 (2) 6292-9988	Facsimile +61 (2) 6292-9977
United Kingdom	Telephone +44 (1454) 777-654	Facsimile +44 (1454) 777-659
United States	Telephone 1800 581-3502	Facsimile 1800 581-3834

While all care has been taken in the development of this product, no loss of any kind is accepted by Protocom Development Systems Ltd or its agents arising from the use of any of these programs. Please read the license agreement before using the software.

Trademarks

Novell and NetWare, ManageWise are registered trademarks of Novell, Inc.
Other trademarks may exist which are not explicitly noted here and remain registered to the appropriate organisations.



Protocom

Table Of Contents

1. INTRODUCTION	4
1.1 OVERVIEW	4
1.2 FEATURE SUMMARY	6
1.3 DEFINITIONS	7
2. SERVER INSTALLATION GUIDE.....	8
2.1 OVERVIEW	8
2.2 INSTALLING THE CONSOLEALERT NLM.....	8
2.3 NETWARE V4.X - UNINSTALLING CONSOLEALERT	10
2.4 LOADING CONSOLEALERT.....	11
2.5 UNLOADING CONSOLEALERT	11
3. WORKSTATION ADMINISTRATION UTILITY INSTALLATION GUIDE.....	12
3.1 OVERVIEW	12
3.2 NWADMIN SNAP-IN UTILITY	12
4. CONSOLEALERT CONFIGURATION.....	14
4.1 OVERVIEW	14
4.2 CONSOLEALERT CONFIGURATION FILE	15
4.3 CONSOLEALERT PROFILE NDS OBJECT	15
4.4 SERVER NDS OBJECT	15
4.5 CONFIGURATION ITEM PRECEDENCE	16
4.6 CONSOLEALERT CONFIGURATION PARAMETERS.....	17
4.7 CONSOLEALERT MESSAGE PARAMETERS	18
5. CONFIGURING SNMP.....	21
5.1 OVERVIEW	21
5.2 FILESERVER SNMP CONFIGURATION	21
5.3 CONFIGURING SNMP ON A MANAGEWISE MANAGEMENT STATION	21
5.4 MIB FILE FORMAT.....	24
6. EVENT LOG.....	26
6.1 OVERVIEW	26
6.2 EVENT TYPES	26
7. COMMONLY ASKED QUESTIONS.....	28
8. CUSTOMER SERVICE AND TECHNICAL SUPPORT.....	29
8.1 CONTACT DETAILS	29
8.2 TECHNICAL SUPPORT	29
9. LICENSE AGREEMENT.....	30



1. Introduction

1.1 Overview

ConsoleAlert provides system administrators with a simple and efficient way to centrally monitor NetWare file servers in a distributed environment. In organizations with a large number of servers, important messages on servers may go unnoticed until too late. ConsoleAlert allows system administrators to respond quickly to issues before users develop operating problems.

ConsoleAlert works by monitoring the Novell server console for user defined messages. Once it detects a message on a server console, it sends an SNMP alert to a centralized network management console, warning network administrators that the message has been detected on the server. Administrators can then proactively respond to the error, potentially preventing data loss or reducing the impact of the network fault.

ConsoleAlert can also be configured to execute console commands in response to particular messages being detected. Thus, minor problems or faults can be automatically corrected, without operator assistance.

ConsoleAlert consists of two major components: a NWAdmin snap-in DLL used to manage the program configuration and the ConsoleAlert NLM that actually monitors a file server console. Figure 1.1-1 shows a sample installation.

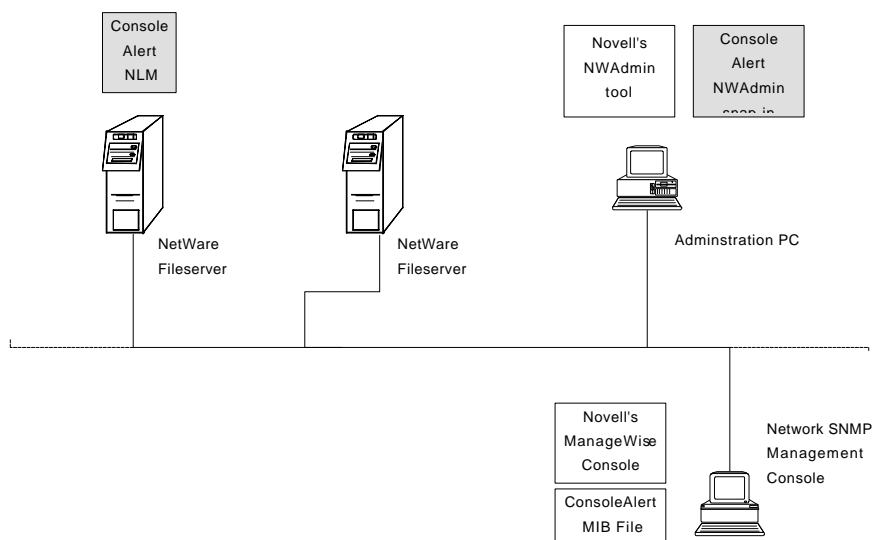


Figure 1.1-1



The first major software component is the ConsoleAlert NLM. This NLM is responsible for actually scanning the file server for the user-specified messages. It works on a server-by-server basis and must be installed on each server that will be monitored. Thus, in the diagram above, the NetWare fileserver on the left has had the ConsoleAlert NLM installed on it and will be monitored, while the server on the right will not be monitored.

When the ConsoleAlert NLM detects a particular console message, it will send out an SNMP alert and optionally execute a console command. The network management station running Novell's ManageWise software, shown at the bottom of the above diagram, will then display the SNMP alert, allowing network administrators to rapidly respond to the problem.

Each installation of ConsoleAlert on a given fileserver has its own set of configuration data, which provides considerable flexibility. This configuration specifies both how the ConsoleAlert program will operate, for example, whether it will write statistics to disk every night, and the set of messages for which it will scan. For example, a dedicated file server may be configured to scan for issues related to disk space, while a server acting as a communications gateway may be configured to scan for messages relating to communications errors.

Each time a ConsoleAlert NLM is loaded, it will attempt to read its configuration information from three separate sources:

- The NDS server object representing the server on which the load took place.
- A configuration file located on the fileserver itself
- A ConsoleAlert profile

The configuration information stored in the NDS server objects and in the c.onfig files is only used by the fileserver with which they are associated. In contrast, profiles can be shared, which provides a powerful mechanism for configuring ConsoleAlert across a large number of fileservers. For example, to add a new message you only need to update a single profile and all file servers using that profile will be automatically configured to detect the new message.

The second major software component on ConsoleAlert is the NWAdmin snap-in DLL. This snap-in module extends Novell's NWAdmin program to allow it to manage the ConsoleAlert configuration information stored in the NDS server and profile objects. In the diagram shown above, the ConsoleAlert snap-in DLL has been installed on the Administration PC.



Protocom

The NWAdmin snap-in module also enables single-point administration, as the configuration of each installation of ConsoleAlert can be controlled from one location, regardless of where the servers physically reside.

1.2 Feature Summary

- Allows networks of any size to centrally report server and network faults
- Any text message can be reported.
- Any number of occurrences can be reported.
- Reporting duration can be varied to requirements.
- For each message, duration and occurrence thresholds can be determined.
- Actions to be implemented are configurable for each message.
- Prioritisation for each action can be established.
- Server based administration enhancement tool.
- NWAdmin compatible administration utility.
- Auditing of user actions, including successful and unsuccessful console login attempts by login name.
- A permanent log file of stats is recorded for use in profiling and trend analysis of system errors.
- Automated execution of console commands when specified conditions are met.

1.3 Definitions

ConsoleAlert Profile Object

ConsoleAlert defines a new NDS object type called a ConsoleAlert Profile Object. This new object can be used by an administrator to setup default configuration and security information. Each NetWare server which has SecureConsole loaded can be configured to use this Console Profile Object to retrieve its configuration and security information. Thus, multiple servers can share the one security and configuration database. The benefit of this approach is that an administrator can make a single change to the shared profile object and multiple servers will be immediately updated.

Messages

ConsoleAlert is used to detect specific text items displayed on the NetWare file server console. Each specific text item for which ConsoleAlert is scanning, for example, the text string "NDS error", is referred to as a message.

NDS Schema

NDS uses a database to store information about the users, file servers and other services contained in your network. The NDS schema defines the structure of this database and ConsoleAlert has extended the schema to support console security definitions. These definitions have been registered and endorsed by Novell.

NetWare Loadable Module (NLM)

ConsoleAlert is a fileserver based application and was written as a Novell NLM. It consumes approximately 100-250k of server memory depending upon the configuration.

NetWare Directory Services (NDS)

ConsoleAlert uses Novell's NDS to store and secure its console security information. NDS is a distributed database which means that parts of the database can exist in different physical locations. A basic understanding of NDS terms and concepts is an important prerequisite for installing and configuring ConsoleAlert.



Protocom

2. Server Installation Guide

2.1 Overview

Installation of ConsoleAlert is accomplished using Novell's INSTALL.NLM on NetWare 4.x and using NWCONFIG.NLM on NetWare 5.x. Note, it is important that these installation programs are used so that they can extend the NDS schema. ConsoleAlert uses the schema definitions created by the install programs to store its configuration information. These extensions have been approved by Novell.

When the download file has been uncompressed, three directories will have been created. The SERVER directory contains the files necessary to install the ConsoleAlert NLM on the NetWare file server. The CLIENT directory contains the files used to install the ConsoleAlert NWAdmin Plug-in module for both Windows 95 and for Windows NT Workstation. The DOC directory contains the program on-line documentation.

Installation of ConsoleAlert is performed in two separate steps:

- Installation of the ConsoleAlert server based application
- Installation of the ConsoleAlert NWAdmin Snap-in for Windows 95 and Windows NT Workstation.

This chapter describes the steps involved in installing the server component, while the next describes how to install NWAdmin Snap-in module.

2.2 Installing the ConsoleAlert NLM

(A) If you are using NetWare Version 4.x, use the command shown below:-

```
PRO_41: LOAD INSTALL
```

If you are using NetWare Version 5.x, using the command:-

```
PRO_50: LOAD NWCONFIG
```

(B) Select the "Product Options" menu (see Figure 2.2-1).



Protocom

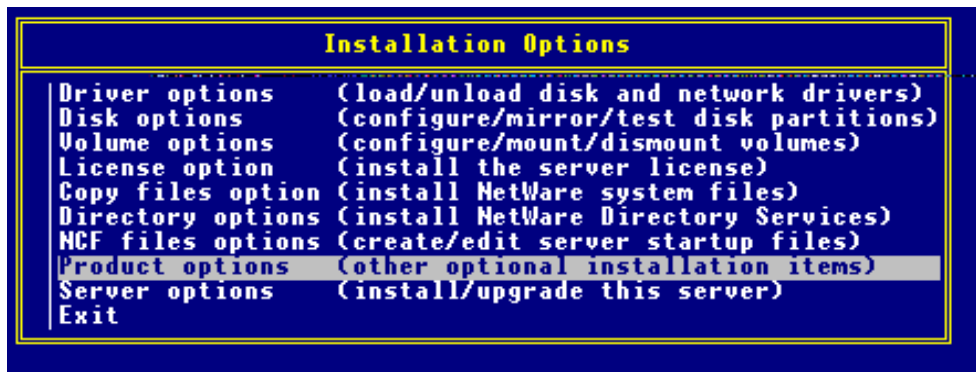


Figure 2.2-1

(C) Select View/Configure/Remove installed products (see Figure 2.2-2).

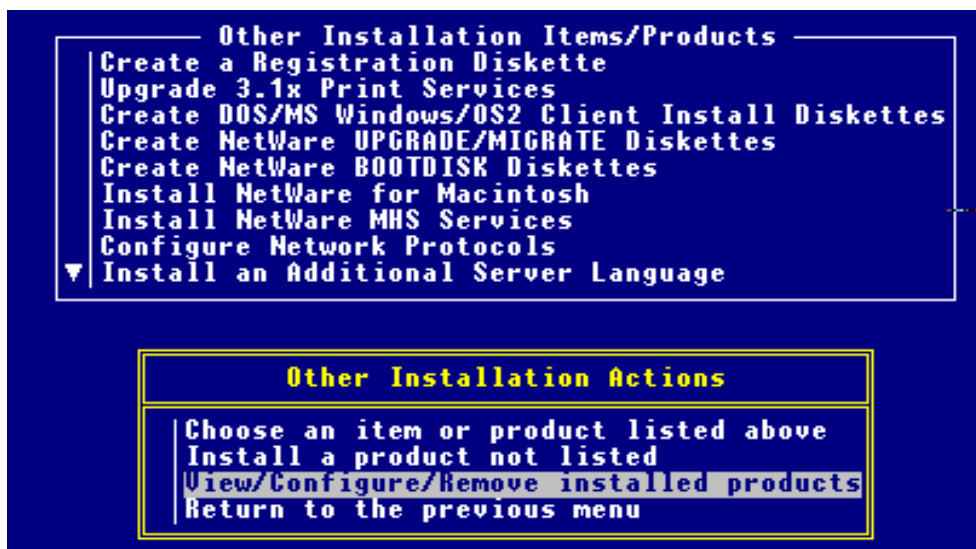


Figure 2.2-2

(D) Press insert and select the path where the ConsoleAlert Installation files can be found. This may be a DOS drive or a valid NetWare path (see Figure 2.2-3).

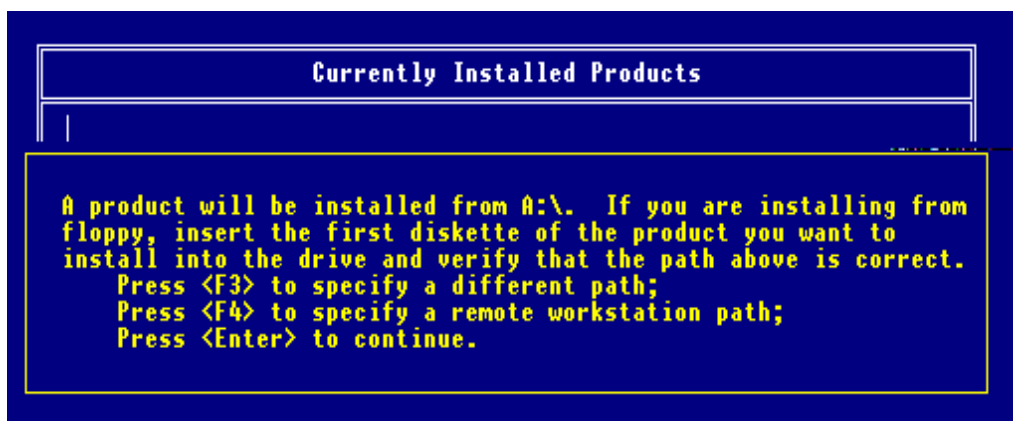


Figure 2.2-3

(E) Select the “ConsoleAlert For NetWare” check box with the space bar or the enter key. If you are upgrading from a previous release of ConsoleAlert please be aware that this process will overwrite your CONALERT.CFG with the defaults for the new version. If you wish to retain your old configuration files, please make a backup of them and restore them after the installation (see Figure 2.2-4).

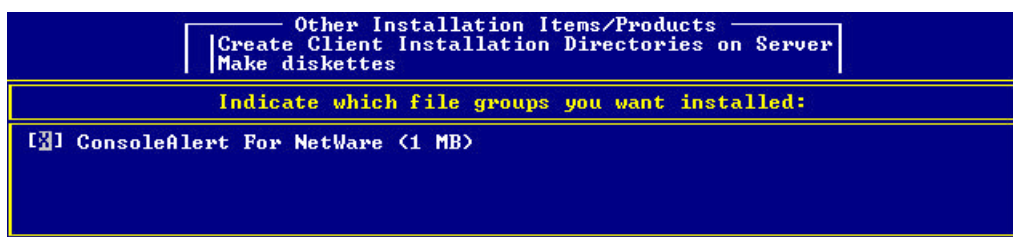


Figure 2.2-4

When ready to install the program files, select the F10 key.

The ConsoleAlert files will be copied to the file server directory SYS:SYSTEM\CONALERT. A search mapping and LOAD command will be automatically added to your AUTOEXEC.NCF file so that ConsoleAlert is started automatically when the fileserver is restarted.

(F) ConsoleAlert has now been successfully installed. The next step is to configure ConsoleAlert to detect the messages that you need to be monitored. See the ConsoleAlert Configuration chapter for a description of the steps involved in this process.

2.3 NetWare v4.x - Uninstalling ConsoleAlert

Select the ConsoleAlert product option from the products screen, press the Delete key and answer “yes” when prompted. (see Figure 2.3-1)



Protocolcom

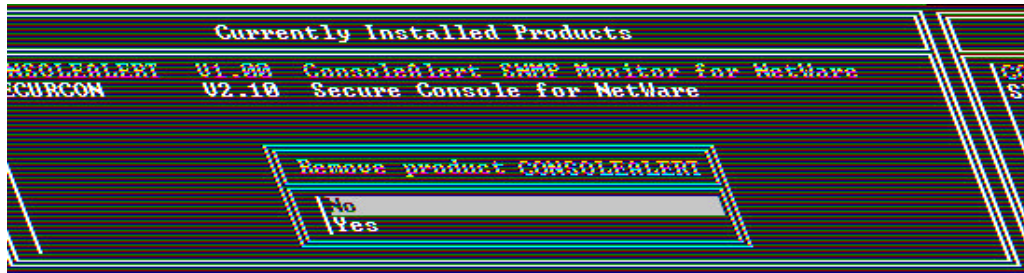


Figure 2.3-1

2.4 Loading ConsoleAlert

If ConsoleAlert is not already loaded, use the following command at the console prompt to activate it :

```
PRO_DEV: LOAD SYS:SYSTEM\CONALERT\CONALERT
```

Note that ConsoleAlert automatically loads as part of the installation process. If the operation fails, check that ConsoleAlert has been properly installed

2.5 Unloading ConsoleAlert

Once loaded, ConsoleAlert can be unloaded from the system console using unload ConsoleAlert, including the full path at the console prompt.

```
PRO_DEV: UNLOAD CONALERT
```

To prevent ConsoleAlert from loading automatically at fileserver reboot, remove the LOAD ConsoleAlert line from the AUTOEXEC.NCF file. ConsoleAlert can be manually invoked using the LOAD ConsoleAlert command.

3. Workstation Administration Utility Installation Guide

3.1 Overview

Once the ConsoleAlert files have been installed on the server, the NWAdmin snap-in Administration utility can be installed onto the appropriate client machines. The Client installation program is found in the CLIENT download directory, in either WIN95 or WINNT directory, depending upon the type of the client machine. In either directory, you should execute SETUP.EXE to begin the installation process.

This windows installation program will automatically install the ConsoleAlert Administration Snap-in Dynamic Linked Library onto the Z:\Public\Win95 directory or the Z:\Public\WINNT directory depending upon client type. The SETUP program will then register the Snap-in with the NWAdmin version appropriate for that machine. If you do not have access to the Z:\PUBLIC\ directory, the install program will simply update the NetWare Administrator registry.

The following instructions assume that you have placed the install files onto a floppy disk. If you have placed them instead onto a NetWork drive, substitute your drive letter for A:\.

3.2 NWAdmin Snap-in Utility

Insert the SecureConsole Diskette into the A: drive or equivalent on the Machine that NWAdmin is installed..

- (A) Select the Startup button, then the Run option.
- (B) At the prompt type A:\WINNT\setup <enter> or A:\WIN95\setup, depending upon the target client configuration.
- (C) The SecureConsole Install wizard (Figure 6.4-8) will then guide you through the installation process.





Figure 3.2-1

If there are any problems with these directories or there are any conflicts when installing, the installation Wizard will prompt you for the corrected information.

4. ConsoleAlert Configuration

4.1 Overview

Once ConsoleAlert has been installed on a server, it needs to be configured so it knows the messages to scan for and the responses it should take once a message has been detected. This configuration information can be stored in the following four configuration items:

- The CONALERT.CFG ConsoleAlert configuration file.
- The ConsoleAlert NDS profile object.
- The NDS server object associated with the file server on which ConsoleAlert is being installed.

To provide for system fault-tolerance and maximum flexibility, each time the ConsoleAlert NLM is loaded on a file server it will attempt to read its instructions from the CONALERT.CFG ConsoleAlert configuration file, a nominated ConsoleAlert NDS profile object and the server NDS object. Each of these configuration items has different characteristics and any combination of the three can be used, so long as ConsoleAlert has access to at least one configuration item.

The configuration file is read first, the profile object second, with the server being read last. Configuration items are assigned a precedence based on the order in which they are read. For example, if a particular program setting is defined in one source and the setting is then subsequently read from another source, the later source carries precedence and will be used by ConsoleAlert.

Before proceeding to configure ConsoleAlert, it is important to have worked out the following issues:

- What console screens you want to monitor.
- The messages that you want to detect on the screens to be monitored.
- The importance of each message and whether it is critical that the message be monitored, even if NDS is unavailable.
- What action is to be taken when a message occurs.
- At what incidence rate do messages warrant action.
- Where are the messages to be reported to.
- When are the messages to be reported.

4.2 ConsoleAlert Configuration File

The first configuration item that ConsoleAlert will attempt to read is CONALERT.CFG configuration file contained in the SYS:\SYSTEM\CONALERT directory. The configuration file carries the least precedence, and thus options specified in the file will be overridden by values contained in either the profile or server object, according to the rules described in [Configuring ConsoleAlert](#).

The advantage of placing a base set of configuration options in the CONALERT.CFG file is that ConsoleAlert will be able to continue to function when Novell's NDS is not available. The disadvantage of using a configuration file is that it is local to the server on which it physically resides, and each server must therefore have their own configuration file, which increases the amount of maintenance.

If you have a number of servers which must be monitored for critical messages, it is recommended that these message definitions be placed into the CONALERT.CFG file. These definitions will be harder to update, but will be used even if NDS is unavailable.

4.3 ConsoleAlert Profile NDS Object

After ConsoleAlert alert has attempted to read the configuration file, it will then attempt to read the nominated profile object, if one has been defined. The use of profiles simplifies the configuration of ConsoleAlert : you only need to establish one profile object that contains detailed specifications of the messages to be detected and all file servers using that profile will share the same configuration information. Profiles thus provide a powerful way to perform program maintenance. For example, to add a new message you only need to update a single profile and all file servers using that profile will be automatically configured to detect the new message.

For simplicity and ease of maintenance it is recommended that a profile be used for each group of file servers that is to monitor similar sets of messages and will have similar response actions defined.

4.4 Server NDS Object

The configuration information contained in the server NDS object will be read last. The configuration contained in this object has the highest precedence, and messages defined at the server level will not have their definitions modified by subsequent reads of other configuration sources. Storing ConsoleAlert configuration information in the server NDS objects combines flexibility, as the information stored in the server object is only relevant for



that server and will not affect others, with ease of management via the NWAdmin snap-in GUI.

It is recommended that any configuration information that needs to be local to a file server be generally placed into the server object for that server. If it is critical that a particular set of messages be monitored, then these definitions should be placed in the server's configuration file.

4.5 Configuration Item Precedence

As the configuration parameters can be stored in the three different locations described above, ConsoleAlert uses strict rules in reading this information. These rules are described below:

- Boolean program options, such as "write statistics to file" will be activated if any one configuration source turns it on. For example, if the CONALERT.CFG file contains the setting `WRITESTATSTOFILE=TRUE`, message detection statistics will be written to the log file regardless of the settings contained in the profile or server object. With boolean options it is thus not possible for an option to be generally enabled within a profile object and then selectively disabled on a server-by-server basis. Program settings processed in this manner are: write statistics to file, use European time, send statistics at midnight and display statistics.
- Text program settings that specify non-boolean values will be taken from the configuration source that has highest precedence, that is, the source that is read last. For example, it is possible to place a generic SNMP MIB Identifier into a profile object, and to then selectively override this generic value by placing different values into specific server objects. Program text settings handled in this way are the system trap number and SNMP MIB Identifier
- ConsoleAlert will scan for all messages defined in the configuration file, profile object and server object. However, it is possible for a generic message definition to be overridden by a more specific message definition having a higher precedence. For example, if an administrator defines a generic message in a profile object, this generic definition can be overridden by a message definition contained within a server object.

The ConsoleAlert SNMP MIB file has a slightly different function to the previous three configuration items. The MIB file is used by network management stations running the ManageWise software to translate the SNMP alerts that are issued by the ConsoleAlert NLM. For an explanation of how this file needs to be used see the Chapter on configuring SNMP.

Each line in the file can contain one parameter. Each of these has the format:

variable_name = value

The following are the default recommendations.

4.6 ConsoleAlert Configuration Parameters

The ConsoleAlert configuration parameters are given below. These parameters control the behaviour of the ConsoleAlert software and can be placed in the config file, server object or profile object.

Each parameter is given two names, the name used in the configuration file (the name following the CFG: in the parameter box) and the field name used in the NWAdmin snap-in module (the name following the NWA).

CFG: SCREENDISPLAY NWA: Display Statistics	Default: True
---	----------------------

By default, ConsoleAlert will display a basic set of detection statistics, including the screen and messages being scanned for, and the number of hourly and daily detections. If this option is set to FALSE, this statistics screen will not be displayed.

Note, if the statistics screen is disabled, ConsoleAlert will continue to detect messages with no other impact upon its functionality.

CFG: WRITESNMPSTATS NWA: Write Statistics To File	Default: True
--	----------------------

If this option is enabled, ConsoleAlert will write detection statistics to the log file every day at midnight. These statistics list the number of daily detections that have taken place for each message.

Writing the statistics to the disk provides a historical record of all message detections that have taken place. This record allows system administrators to keep track of the system events that occur on their NetWare file servers.

CFG: SENDDAILYSTATS NWA: Send statistics at midnight	Default: True
---	----------------------

This option is not currently supported.



CFG: EUROPEANTIME NWA: Use European time	Default: True
---	----------------------

By default, ConsoleAlert will use the European format (dd/mm/yy) when displaying dates on the screen and in the log file. If this option is set to FALSE, US date format will be used instead (mm/dd/yy).

CFG: PROFILE NWA: Configuration Profile	Default: NONE
--	----------------------

ConsoleAlert is capable of reading configuration information from the config file, server NDS object and a profile NDS object. The profile parameter, which is only contained in either the config file or the server object specifies the profile to use. Note, if a profile is defined in both the configuration file and the server object, the definition contained in the server object will take precedence.

CFG: MIBID NWA: MIB Identifier	Default: 1.3.6.1.4.1.23.2.8897
---	---------------------------------------

ConsoleAlert uses SNMP to send alerts to the network management console. The MIBID parameter specifies the MIB to use in decoding the ConsoleAlert SNMP alarms. This value must match the value contained in the ConsoleAlert MIB compiled into Novell's ManageWise product.

CFG: CASYSTRAP NWA: MIB Identifier	Default: 5
---	-------------------

If ConsoleAlert experiences problems during execution, for example, if it loses read rights to the server object, it will issue an SNMP trap to inform network administrators. This parameter specifies the SNMP trap number used by ConsoleAlert for these system errors.

4.7 ConsoleAlert Message Parameters

The message parameters define the set of messages for which ConsoleAlert will scan. As with the program parameters documented above, the message parameters can be contained within the config file, profile object and server object.

The parameters given below are used for each message, all parameters are required and there are no defaults.



CFG: SCREENNAME NWA: Screen Name	Default: No default
---	----------------------------

The name of the screen that ConsoleAlert should scan in attempting to find the message. Note, the screen name must match the NLM screen name exactly.

CFG: MESSAGETEXT NWA: Message	Default: No default
--	----------------------------

The message text for which ConsoleAlert will scan. The matching performed is case-insensitive and all white space is significant.

CFG: MESSAGETRAPNUM NWA: Trap Num	Default: No default
--	----------------------------

The trap parameter specifies the SNMP trap number that will be used by ConsoleAlert when it issues an SNMP trap for the selected message. The ConsoleAlert MIB contained on the network management station defines the valid set of SNMP alerts that can be used by ConsoleAlert.

To simplify the management of ConsoleAlert SNMP alerts it is a good idea to group the messages being detected into categories, for example, high, medium and low priority, and assign each category a different SNMP trap number. This will allow messages appearing on an SNMP management console to be easily categorized according to the severity of the error.

CFG: MESSAGECOMMAND NWA: Command	Default: No default
---	----------------------------

The system console command to execute once the message has been detected.

CFG: MESSAGECAPTURELINES NWA: Lines	Default: No default
--	----------------------------

This option is not currently supported.

CFG: MESSAGETHRESHOLD NWA: Threshold	Default: 60
---	--------------------

The threshold defines the number of times that a message must be detected during a given time period (see the period parameter below) before ConsoleAlert will send out an SNMP trap or execute a console command.



The threshold value represents the number of detections that must take place before any action is taken. Thus a threshold of one will result in every detection triggering a response from ConAlert; a threshold of two means that every second detection will trigger a ConAlert response (so long as both detections take place within the specified period).

CFG: MESSAGEPERIOD NWA: Period	Default: 60
---	--------------------

This parameter specifies the number of minutes during which the message detections will count towards reaching the threshold level. After this period has expired, if the number of message detections has not reached the required threshold, the error detection count is reset to zero. For example, if a message had a threshold of 5 and a period of 60 minutes and only three detections were made during the first hour, then during the second hour five messages would need to be detected, not two.



5. Configuring SNMP

5.1 Overview

Novell's ManageWise software is a powerful tool that lets network administrators monitor the SNMP traps generated by various nodes within a network.

For the SNMP component of ConsoleAlert to work correctly, all file servers using ConsoleAlert, the network management station and the ConsoleAlert configuration information (contained in the config file, server and profile objects) must be properly synchronized.

5.2 Fileserver SNMP configuration

To configure your file servers to send the ConsoleAlert SNMP alerts (along with alerts from other applications) the text file TRAPTARG.CFG in the SYS:ETC\ directory must include the destination address of the SNMP management station that is to receive the SNMP alerts.

NetWare supports the transmission of SNMP alerts via IPX or TCP/IP and is controlled by indicating appropriate addresses or host names in the TRAPTARG.CFG file.

Note, the SNMP.NLM must be unloaded and reloaded for any changes made to the TRAPTARG.CFG file to take effect.

5.3 Configuring SNMP on a ManageWise Management Station

ManageWise is Novell's management console used to monitor most aspects of an organisation's network. The main screen of ManageWise is shown in Figure 5.3-1.



Protocom

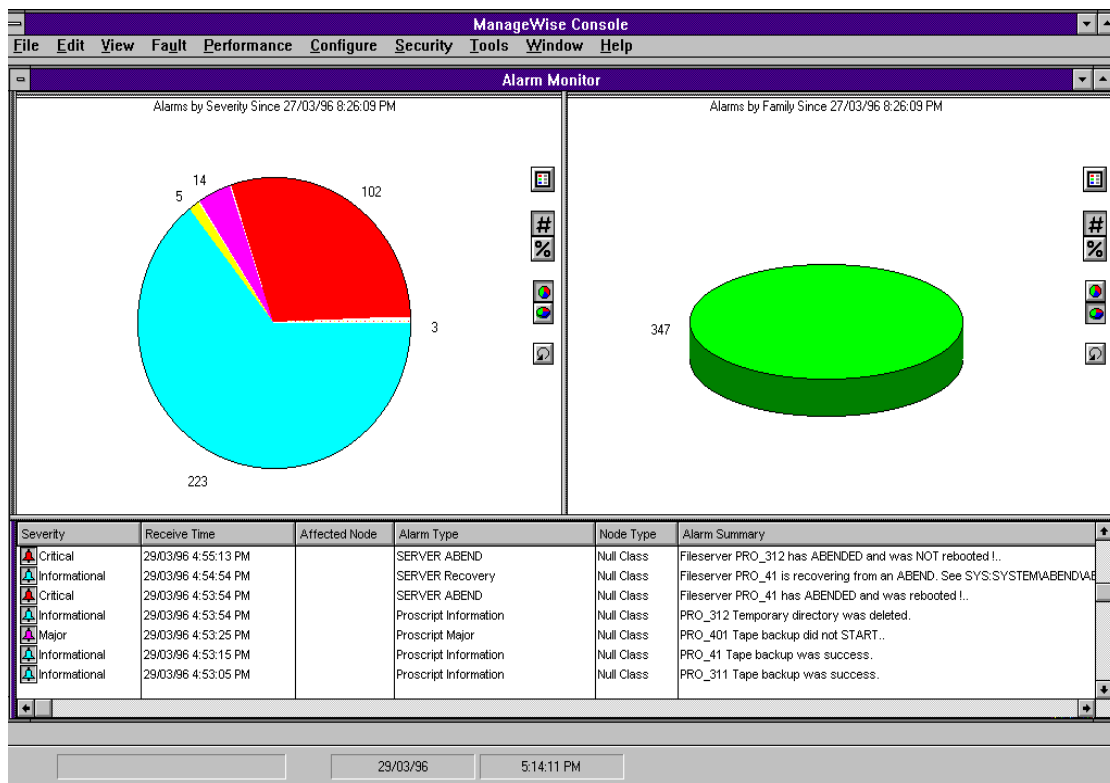


Figure 5.3-1

ConsoleAlert communicates with this console using the industry SNMP protocol. In order for Managewise to interpret the console alert messages (known as SNMP traps) the ConsoleAlert management information bulletin (MIB) definition file must be compiled using ManageWise. This is achieved by using the following process:-

- Copy the ConsoleAlert.MIB text file to the <drive>\MW\NMS\SNMPMIBS\CURRENT directory where <drive> is the location of ManageWise.
- From the ManageWise console, select TOOLS, COMPILE MIBS and COMPILE.

See Figure 5.3-2 below.

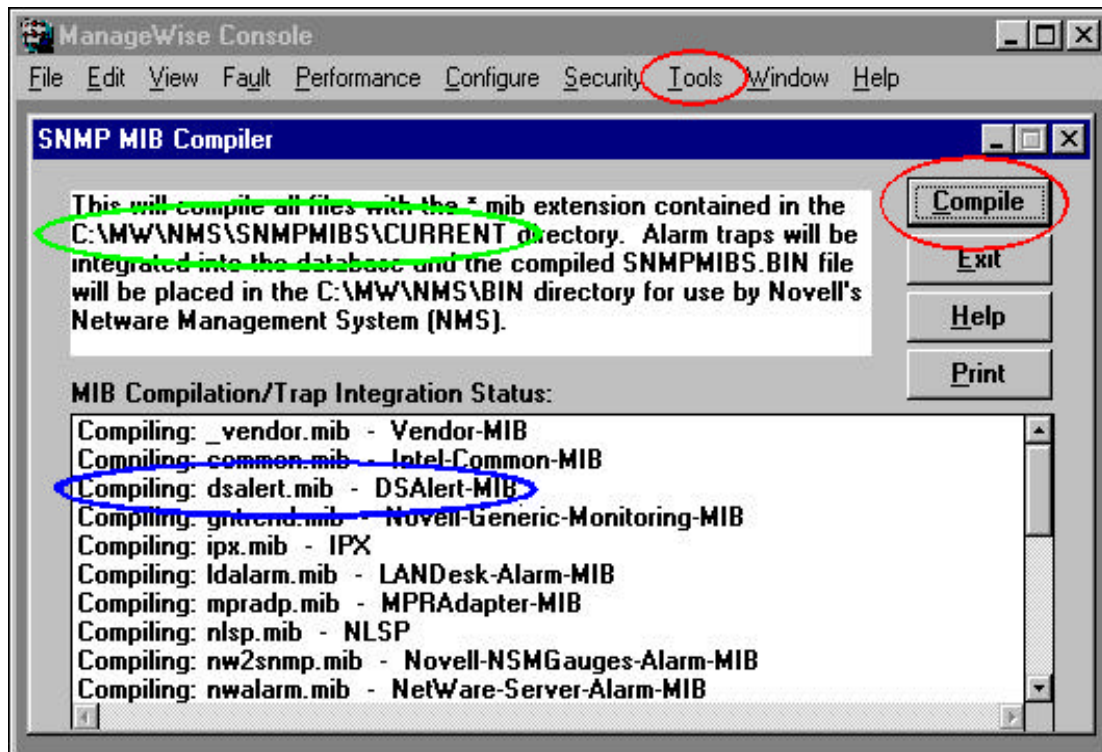
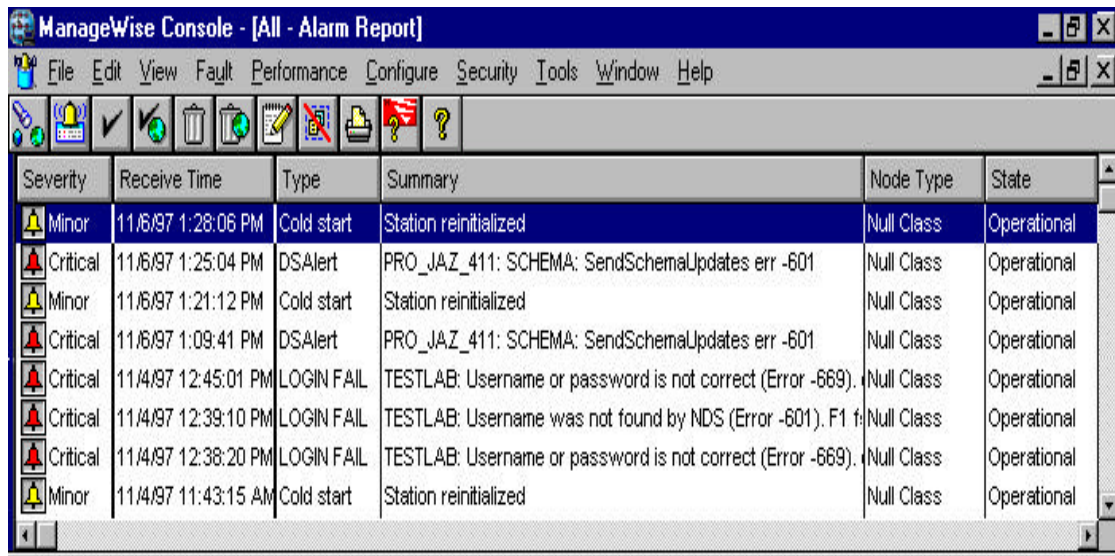


Figure 5.3-2

Hint: If the MIB file is ever changed, it must be recompiled by ManageWise before it will take effect. It is not sufficient to simply change the MIB file.

If no errors occurred during the compilation ManageWise is ready to accept console alerts from ConsoleAlert.

When ManageWise receives alerts from ConsoleAlert, ManageWise displays the alerts in the alarm monitor, as shown in Figure 5.3-3.



Severity	Receive Time	Type	Summary	Node Type	State
Minor	11/6/97 1:28:06 PM	Cold start	Station reinitialized	Null Class	Operational
Critical	11/6/97 1:25:04 PM	DSAlert	PRO_JAZ_411: SCHEMA: SendSchemaUpdates err -601	Null Class	Operational
Minor	11/6/97 1:21:12 PM	Cold start	Station reinitialized	Null Class	Operational
Critical	11/6/97 1:09:41 PM	DSAlert	PRO_JAZ_411: SCHEMA: SendSchemaUpdates err -601	Null Class	Operational
Critical	11/4/97 12:45:01 PM	LOGIN FAIL	TESTLAB: Username or password is not correct (Error -669).	Null Class	Operational
Critical	11/4/97 12:39:10 PM	LOGIN FAIL	TESTLAB: Username was not found by NDS (Error -601). F1 f	Null Class	Operational
Critical	11/4/97 12:38:20 PM	LOGIN FAIL	TESTLAB: Username or password is not correct (Error -669).	Null Class	Operational
Minor	11/4/97 11:43:15 AM	Cold start	Station reinitialized	Null Class	Operational

Figure 5.3-3

5.4 MIB File Format

The purpose of the MIB file is to translate the raw SNMP alerts into meaningful information that can be understood by operators. Thus, when ConsoleAlert detects a message and sends an SNMP alert, it is the MIB file that will determine how ManageWise displays the alert.

An example of a typical entry, setup by a administrator could be;

```

ConsoleAlertMinor TRAP-TYPE
    ENTERPRISE ConsoleAlert
    DESCRIPTION
        "ConsoleAlert Minor Event."
    --#TYPE "ConsoleAlert"
    --#SUMMARY "%S."
    --#ARGUMENTS {0}
    --#SEVERITY NONCRITICAL
    --#TIMEINDEX 1
    --#HELP ""
    --#HELPTAG 0
    --#STATE OPERATIONAL
    ::= 4
  
```




6. Event Log

6.1 Overview

The event log is used to keep a record of the processing performed by ConsoleAlert. The log file is always given the name CONALERT.LOG and will be stored in the SYS:\SYSTEM\CONALERT directory.

Each line with the log represents one event and contains :

- date and time event occurred
- type of event
- name of user logged onto the console when the event occurred
- extract information pertinent to the event

6.2 Event Types

The events can be the following types:

Event Type	Event Description
START	The ConsoleAlert NLM has begun execution. The time recorded with this event indicates when ConsoleAlert will begin scanning the server screens.
END	The ConsoleAlert NLM has been unloaded and is no longer monitoring the server. This event is written if the NLM is unloaded, or if there is a system fault and ConsoleAlert has shut itself down, for example, if there was no available server memory.
INFO	This event is used to provide administrators with information about the state of processing within ConsoleAlert. INFO event records will be written when: <ol style="list-style-type: none"> 1. ConsoleAlert is first loaded. At this time an INFO record will be written for each configuration source noting whether the config source is accessible or not. 2. ConsoleAlert rereads its configuration sources due to user config updates. 3. ConsoleAlert successfully connects to a screen that it has been configured to monitor. 4. ConsoleAlert loses its connection (or fails to establish one) to a screen that it has been configured to monitor.
MSG_FOUND	Each time ConsoleAlert detects a message, it writes a MSG_FOUND event into the event log. The event log thus contains a complete trace of all events detected.
MSG_CMD_EXC	If ConsoleAlert has been configured to execute a console command in response to detecting a message, it will append an event log record before executing the command.

WARNING	<p>Warning events inform network administrators that an issue has arisen that may affect ConsoleAlert's performance. Depending upon when a warning event is trapped, ConsoleAlert will continue to either operate or will continue to load (if the event was trapped while the ConsoleAlert NLM was loading).</p> <p>Events that cause warning log records to be written are if unknown or badly formatted program options are read from a configuration source (config file, profile object or server object). Warning events are also written if no messages at all are read from a configuration source.</p> <p>All warning events detected by ConsoleAlert will cause a warning message to be printed to the screen. However, no SNMP alert will be sent.</p>
ERROR	<p>Error events inform network administrators that a condition has arisen that degrades the performance of ConsoleAlert in a specific manner. If error events arise during loading the NLM load processing, ConsoleAlert will automatically terminate itself. If an error event occurs after ConsoleAlert has been successfully loaded, ConsoleAlert will continue to execute, though with reduced functionality.</p> <p>Error events are triggered if the two required parameters (the MIB identifier and the ConsoleAlert system trap number) have not been defined in any configuration source or if a config source that was previously readable becomes unreadable.</p> <p>All error conditions detected after the MIB identifier and system trap number have been read in will cause ConsoleAlert to send an SNMP alert to the network management console and print an error message on the system console.</p>
FATAL	<p>Fatal events are triggered when the file server environment prevents ConsoleAlert from continuing to process. In such a situation, the ConsoleAlert NLM has no choice but to inform network administrators of the condition and to terminate further processing.</p> <p>Any fatal event detected will cause ConsoleAlert to send an SNMP alert to the network management console (so long as the MIB identifier and system trap number have been read) and print an error message on the system console.</p>

7. Commonly Asked Questions

1. Q. What performance overhead does the product put on the server?

A. The total space required by the ConsoleAlert installation is 563kb, of this the NLM is only 114kb, the log file that is generated has a maximum size of 2mb. Processor useage time is kept to a minimum as ConsoleAlert does not use a polling method to test the Directory Services screen to ascertain when an error message has been generated. This maximises server work rates.

2. Q. Can the Message be sent to mutiple consoles?

A. Yes, any number of consoles can be included as destinations for the alert mesages. This allows diferent offices or branches to be informed of error conditions simultaneously, simplifying network administration and maintenance.

3. Q. Can we set the software to page our support people when fatal Messages occur?

A. ConsoleAlert and Managewise can be configured to send pager messages directly to your service personnel immediately a fatal NDS error condition occurs. This allows technicians to be proactive in correcting the problem before users can become disgruntled and down time becomes excessively costly.

8. Customer Service and Technical Support

8.1 Contact Details

USA

Telephone 1800- 5813502
 Facsimile 1800- 5813834
 sales@serversystems.com
 support@serversystems.com

Australia

Telephone +61 (2) 6292-9988
 Fax back +61 1800-581-3834
 Facsimile +61 (2) 6292-9977
 sales@serversystems.com
 support@serversystems.com

United Kingdom

Telephone +44 1454-777-654
 Facsimile +44 1454-777-659
 sales@serversystems.com
 support@serversystems.com

8.2 Technical Support

Technical support is available to all registered users.

CompuServe

Mail your technical support questions to 100033,3202

Internet

Mail your technical support questions to
 support@serversystems.com or
 100033.3202@compuserve.com

WWW

<http://www.serversystems.com>

Facsimile

United States 1800 581-3834
 Australia +61 (2) 6292-9977
 United Kingdom +44 1454 777 659



9. LICENSE AGREEMENT

IMPORTANT- READ CAREFULLY BEFORE USING THIS SOFTWARE.

By using the software or opening the sealed packet containing the software, you agree to the following software license agreement.

LICENSE AGREEMENT

This is a legal agreement between you (either an individual or an entity) and Protocom Development Systems Pty Ltd. By using the software or opening the sealed software packet you are agreeing to be bound by the terms of this agreement. If you do not agree to the terms of this agreement, promptly return the unopened software packet and the accompanying items (including written materials and binders or other containers) to the place you obtained them for a refund.

1. **GRANT OF LICENSE.** This license Agreement permits you to use one copy of the enclosed software program (the "SOFTWARE") on a single computer. The SOFTWARE is in use on a computer when it is loaded into temporary memory (i.e. RAM) or installed into permanent memory (e.g. hard disk, CD-ROM or other storage devices) of that computer. However, installation on a network server for the sole purpose of internal distribution shall not constitute "use" for which a separate license is required, provided you have a separate license for each computer to which the SOFTWARE is distributed.
2. **COPYRIGHT.** The SOFTWARE is owned by Protocom Development Systems Pty Ltd. and is protected by international copyright laws and treaty provisions. Therefore, you must treat the SOFTWARE like any other copyrighted material (e.g. a book or musical recording) except that you may either (a) make one copy of the SOFTWARE solely for backup or archival purposes, or (b) transfer the SOFTWARE to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials accompanying the SOFTWARE.
3. **OTHER RESTRICTIONS.** You may not rent or lease the SOFTWARE, but you may transfer the SOFTWARE and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. You may not reverse engineer, decompile, or disassemble the SOFTWARE. If the SOFTWARE is an update or has been updated, any transfer must include the most recent update and all prior versions.
4. **DUAL MEDIA SOFTWARE.** If the SOFTWARE package contains both 3.5" and 5.25" disks, then you may use only the disks appropriate for your single



user computer. You may not use any other disks on another computer or loan, rent, lease or transfer them to another user except as part of a permanent transfer (as provided above) of all SOFTWARE and written materials.

LIMITED WARRANTY

1. MEDIA. Protocom Development Systems Pty Ltd. warrants that the physical media that the SOFTWARE is supplied on, against physical defects for a period of ninety (90) days from the date of purchase. Your sole remedy for defective media is replacement.

2. SOFTWARE. Protocom Development Systems Pty Ltd. warrants that if the SOFTWARE fails to substantially conform to the specifications in the documentation accompanying the SOFTWARE and if the nonconformity is reported in writing by you to Protocom Development Systems Pty Ltd. within ninety (90) days from the date that the software is received, then Protocom will either remedy the nonconformity or offer to refund the license fees paid by you upon return of all copies of the SOFTWARE and documentation to Protocom.

3. DISCLAIMER. Protocom Development Systems Pty Ltd. ("Protocom") makes no warranty, representation or promise not expressly set forth in this limited warranty, Protocom disclaims and excludes any and all implied warranties of title and fitness for a particular purpose. Protocom does not warrant that the SOFTWARE or associated documentation will satisfy your requirements or that the software or documentation are without defect or error or that the operation of the software will be uninterrupted. Some countries do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. This warranty gives you specific legal rights which vary from country to country.

4. LIMITATION ON LIABILITY. Protocom's aggregate liability arising from or relating to your use of the SOFTWARE, associated documentation is limited to the total of all payments made by or for you for this copy of the SOFTWARE and documentation. Neither Protocom Development Systems nor any of its licensors, employees or agents shall in any case be liable for any special, incidental, consequential, indirect or punitive damages even if advised of the possibility of those damages. Neither Protocom nor any of its licensors, employees or agents is responsible for lost profits or revenue, loss of use of the SOFTWARE, loss of data, costs of recreating lost data, the costs of any substitute equipment or program. Some countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

