

# Integrity Desktop Manager Guide

Managing Integrity Desktop in the Enterprise



Smarter Security™

ZLD 1-0220-0400-2003-07-29

This document is the *Integrity Desktop Management Guide*, product version 4.0.

## Document Revision History

Zone Labs Document Number	Document Publication Date	Comments
ZLD 1-0220-0400-2003-07-29	2003-07-29	Document for Integrity Desktop, product version 4.0.

## About Zone Labs, Inc.

Zone Labs, one of the most trusted brands in Internet security, is a leading creator of endpoint security solutions protecting millions of PCs from risks posed by hackers and data theft. The company's award-winning endpoint security product line is deployed in global enterprises, small businesses and consumers' homes, protecting them from Internet-borne threats. Zone Labs Integrity™ is an endpoint security management platform that protects corporate data and productivity while ZoneAlarm, ZoneAlarm Pro and ZoneAlarm Plus are among the most popular and successful Internet security products available today.

Founded in 1997, Zone Labs is a private company headquartered in San Francisco, California, USA, with European headquarters in Frankfurt, Germany. For more information, please visit Zone Labs at [www.zonelabs.com](http://www.zonelabs.com).

---

© 2002 Zone Labs, Inc. All Rights Reserved. TrueVector, ZoneAlarm, Zone Labs, the Zone Labs logo and Zone Labs Integrity are either registered trademarks or trademarks of Zone Labs, Inc. U.S. Patent No. 5,987,611. Reg. U.S. Pat. & TM Off. Cooperative Enforcement and Policy Lifecycle Management are service marks of Zone Labs, Inc. All other trademarks are the property of their respective owners. 1060 Howard Street, San Francisco, CA 94103. USA.

No part of this publication may be reproduced, distributed or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Zone Labs, Inc. and its licensors. Users Subject to Standard License Terms and Conditions. Zone Labs, ZoneAlarm, ZoneAlarm Pro, True Vector, and Zone Labs Integrity are trademarks of Zone Labs, Inc. All other trademarks are the property of their respective owners.

You may not export or re-export this publication outside of the jurisdiction in which you obtained it without the appropriate United States or foreign government licenses. You may not reverse engineer, de-compile, or disassemble, modify, or create derivative works based upon, this publication in whole or in part or remove any proprietary or other legal notices or labels on this publication.

Information in this publication is subject to change without notice. The publication may have errors or defects and its accuracy and reliability are not guaranteed. ACCORDINGLY, ZONE LABS MAKES NO EXPRESS WARRANTIES, ORAL OR WRITTEN, REGARDING THE PUBLICATION. THE PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. ZONE LABS DISCLAIMS ANY AND ALL OTHER WARRANTIES, WHETHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

This product includes software developed by the Apache Software Foundation <http://www.apache.org>.

## Chapter 1

### Introducing Integrity Desktop 4.0

---

<b>What's New in Integrity Desktop 4.0?</b> .....	1
<b>Integrity Desktop Benefits</b> .....	2
<b>Running Integrity Desktop</b> .....	3
Periodic Configuration File Downloading .....	3
Automatic Alert Log Uploading .....	3
Single Policy Enforcement .....	3
Cooperative Enforcement .....	3
<b>How this Guide is Organized</b> .....	3
<b>Naming Conventions Used in this Document</b> .....	4
<b>Syntactic Conventions Used in this Document</b> .....	5
Line Breaks and Hyphenation .....	5
Single-step Procedures .....	6
Specifying Configuration File or Policy File Parameters .....	6
<b>Notes and Cautions</b> .....	6
<b>Single-Step Procedures</b> .....	7

## Chapter 2

### System Requirements

---

The following sections list the hardware and software components required to successfully install and run the Integrity Desktop version 4.0 program. ....	8
<b>Recommended System Requirements</b> .....	8
<b>Minimum System Requirements</b> .....	8
<b>Windows System Requirements</b> .....	8

## Chapter 3

### Upgrading to Integrity Desktop 4.0

---

<b>Upgrade Guidelines</b> .....	9
Upgrading, SMS, and Rebooting .....	10
Upgrading and Policy Deployment .....	10
<b>Specific Upgrade Scenarios</b> .....	10
Upgrading ZoneAlarm Pro 2.6 to Integrity Desktop 4.0 .....	10
Using the Reset Command Line Parameter .....	11
Upgrading Zone Alarm Pro - Integrity Desktop 3.1 to Integrity Desktop 4.0 .....	11
Upgrading ZoneAlarm Pro Users to Integrity Desktop 4.0 .....	12

---

# Chapter 4

## Installing Integrity Desktop 4.0

---

<b>Before You Begin</b> .....	13
Initial Installation, SMS, and Reboot .....	13
<b>Performing a Default Installation</b> .....	13
Default Installation .....	14
<b>Performing a Non-default Installation</b> .....	14
<b>Installation Command Lines</b> .....	14
Overview of Installation Command Lines .....	15
Limitations on Installation Command Line Length .....	15
Installation Command Line Switches .....	16
General Installation Command Line Switches .....	17
Tutorial and Wizard Installation Command Line Switches .....	27
Set or Modify Password Installation Command Line Switches .....	28
The Configuration File Installation Command Line Specifier .....	33
The Policy File Installation Command Line Switch .....	33
<b>Post-installation Configuration Files</b> .....	34
<b>Optional Installation Resources</b> .....	34
Optional Installation Wrappers .....	34
Optional Installation Deployment Tools .....	34
<b>Changing an Existing Installation</b> .....	34

# Chapter 5

## Using Operational Command Lines

---

<b>Types of Command Lines</b> .....	35
Types of Command Lines .....	35
<b>Operational Command Lines</b> .....	36
Overview of Operational Command Lines .....	36
Operational Command Line Switches .....	36
Set or Change License Key Operational Command Line Switch .....	37
Set or Modify Passwords Operational Command Line Switches .....	38
The -config Operational Command Line Switch .....	41
The Policy Operational Command Line Switch .....	41
<b>Creating and Running Operational Command Lines</b> .....	41
Creating an Operational Command Line Batch File .....	42
Running an Operational Command Line Batch File .....	42

# Chapter 6

## Saving Configuration Files

---

<b>Saving Integrity Desktop Settings</b> .....	44
--	----

---

# Chapter 7

## Uninstalling Integrity Desktop

---

<b>Before You Begin</b> .....	46
<b>Performing Prompted Uninstallation</b> .....	46
Default Uninstallation .....	46
<b>Performing a Command Line Uninstallation</b> .....	47
<b>Upgrading to Integrity Desktop</b> .....	47

# Chapter 8

## VPN Setup

---

<b>Before You Begin</b> .....	48
Supported Versions of Microsoft Windows .....	48
Supported VPN Protocols .....	48
<b>Overview of VPN Setup</b> .....	49
<b>Identifying Trusted Network Resources</b> .....	49
Types of Network Resources .....	50
<b>Granting Programs Access</b> .....	53
Granting Program Access in Response to an Alert Box .....	54
Granting Program Access with the Control Center .....	54
<b>Enabling VPN Protocols</b> .....	56
<b>Using Troubleshooting Aids</b> .....	57
Enabling Alert Logging .....	58
Automatically Assigning New Network Resources to the Trusted Zone .....	60
Enabling Program Learning Mode .....	60

# Chapter 9

## Automatically Downloading Configuration Files

---

<b>What's New in this Release?</b> .....	62
<b>Downloading Updated Configuration Files</b> .....	62
Before You Begin .....	62
Web Server Requirements .....	63
Configuration or XML Policy File Requirements .....	63
<b>Configuring the <i>autoconfig</i> Element or Section</b> .....	63
Editing the <i>autoconfig</i> Element in an XML Policy File .....	63
Editing the <i>autoconfig</i> Section in a Configuration File .....	66
<b>Updating with Zone Labs Integrity Server</b> .....	69

---

# Chapter 10

## Automatically Uploading Alert Logs

---

<b>What's New in this Release?</b> .....	70
<b>Uploading Archived Alert Logs</b> .....	70
Before You Begin .....	70
Microsoft IIS.....	71
ZALogUpload Active Server Page .....	72
<b>Configuring the <i>autouploadlog</i> Element or Section</b> .....	72
Managing Microsoft IIS Security.....	72
Editing the <i>autouploadlog</i> Element in an XML Policy File .....	77
Editing the <i>autouploadlog</i> Section in a Configuration File .....	80
<b>The TempUploadResponseLogFile Receipt File</b> .....	82
<b>Responding to Certificate Alert Dialog Boxes</b> .....	83
<b>Locating and Viewing Log Files</b> .....	84
Uploaded Log Files Naming Convention .....	84
Example Log File Contents .....	84
Reading Uploaded Alerts .....	85
Event Types.....	86
ICMP Message Types.....	88
TCP Packet Type Flags.....	89

# Chapter A

## Supplemental Administrator Utilities

---

<b>Running the Product Finder Utility</b> .....	90
<b>Using the Policy Update Utility</b> .....	91
Policy Update Utility File-naming Conventions .....	91
Running the Policy Update Utility.....	93

# Appendix B

## Managing Passwords and Files

---

<b>Managing Clear-text Passwords</b> .....	94
Using Windows Script Files .....	94
Managing Custom Configuration Files .....	95
<b>Managing Clear-text Data Files</b> .....	96
Managing Clear-text Policy Files.....	96
Managing Clear-text Alerts and Log Files.....	96

# Index

---

---

## Introducing Integrity Desktop 4.0

---

This document describes how to install and configure Integrity Desktop 4.0.

Integrity Desktop 4.0 is designed to address the most rigorous of network security challenges posed by existing and emerging hostile threats on the Internet. This includes targeted as well as random intrusions such as port scanning, operating system fingerprinting, denial of service attacks as well as the full array of malware threats including Trojan horses, worms, viruses and malicious code.

The Integrity Desktop security engine does not rely on signature updates such as those used by most anti-virus software and intrusion detection systems. Instead, Integrity Desktop utilizes advanced application control and sophisticated protection at the network layer to neutralize threats.

### What's New in Integrity Desktop 4.0?

The following summarizes some of the more notable additions or enhancements to Integrity Desktop 4.0:

- Custom Security Zones

Specify endpoint firewall rule sets for traffic to or from different segments of trusted networks

Value: lets administrators assign distinct levels of access to different network resources

- Location-based Rules

Apply firewall and application rules based on the MAC addresses of network resources

Value: customize access to devices at particular locations, even when they use non-unique private IP addresses

- Total Client Lockdown

Can now prevent all users - even those with local administrator rights on their PCs - from modifying IT-specified policies or disabling Integrity protections.

Value: ensures endpoint security is in force for all users at all times

- User Spoofing Protection

Prevent simulated keyboard or mouse input designed to disable endpoint security

Value: stops a threat other endpoint firewall products are susceptible to

- **Outbound E-mail Protection**

Stop hacker code from sending E-mail or Spam using an employee's personal E-mail account

Value: defeats a potential security breach and source of enterprise embarrassment or liability
- **Customizable User Interface**

Modify overview and tech support text, program permissions display, and other UI elements

Value: lets administrators guide end users' experiences based on the needs of the enterprise
- **Automatic VPN Detection and Configuration**

Apply appropriate settings the first time a user attempts a remote access connection

Value: frees end users from making VPN configuration decisions and enables trouble free remote access
- **Incremental Policy Updates**

Push only changed policy attributes to clients, rather than entire new policy files

Value: minimizes bandwidth usage and provides a policy history that aids troubleshooting

## Integrity Desktop Benefits

Integrity Desktop 4.0:

- Provides the most proven and relied-upon endpoint security protection available
- Closes a huge hole in network security: Trojan Horses exploiting unprotected PCs
- Defends against custom-coded attacks designed to steal enterprise data
- Blocks new forms of attack that defeat legacy products such as anti-virus and Intrusion Detection Systems with a superior “guilty until proven innocent” approach
- Adds immediate security enhancement to any computer upon installation
- Thwarts the most sophisticated new hacker tactics, such as abuse of trusted applications
- Quarantines potentially dangerous e-mail attachments that evade other defenses
- Provides a simple method for upgrading, via a configuration change, to a centrally managed client
- Includes support for Windows 95

Integrity Desktop 4.0 is recognized by numerous reviewers as a superior security effectiveness and usability. Benefit from using Zone Labs, the most trusted name in endpoint security.

# Running Integrity Desktop

Integrity Desktop is designed to operate in two types of networks:

- Networks that do not contain any instance of Integrity Server
- Hybrid networks that contain both a Web server and an Integrity Server, and a mix of Integrity Desktop, Integrity Flex, and Integrity Agent clients

In hybrid networks, Integrity Desktop operates in conjunction with a Web server while Integrity Agent and Integrity Flex operate in conjunction with the network's Integrity Server.

The following sections provide an overview of various aspects of Integrity Desktop operation.

## Periodic Configuration File Downloading

When used in conjunction with a standards-based Web server, Integrity Desktop can be configured to periodically download ("pull") security policies from the server.

This differs from the deployment ("push") of enterprise security policies performed in a network equipped with Integrity Server.

## Automatic Alert Log Uploading

If a Microsoft Internet Information Services (IIS) server is available on the local network, Integrity Desktop can be configured to periodically upload archived alert log files to the server.

## Single Policy Enforcement

Integrity Desktop recognizes only a single security policy operating on the client computer.

This differs from the policy arbitration performed when Integrity Agent or Integrity Flex operates in a network equipped with Integrity Server.

## Cooperative Enforcement

This feature enables administrators to specify the acceptable version level of Integrity Desktop when a user logs on through an Integrity-compatible Cisco VPN. See the *Administrator Guide* for more detailed information about co-operative enforcement.

## How this Guide is Organized

This management guide explains how to install, configure, and manage Integrity Desktop. The remaining chapters in this document contain the following information:

- Chapter 2, "System Requirements," provides an overview of the hardware and software necessary to run Integrity Desktop.

- Chapter 3, "Upgrading to Integrity Desktop 4.0," describes how to upgrade an existing Zone Labs product to Integrity Desktop 4.0.
- Chapter 4, "Installing Integrity Desktop 4.0," describes how to perform an initial installation of Integrity Desktop.
- Chapter 5, "Using Operational Command Lines," describes how to use command lines to modify user-level or installation-level passwords, or to cause Integrity Desktop to read new settings from a policy file.
- Chapter 6, "Saving Configuration Files," explains how to save Integrity Desktop's settings into a text-based policy file.
- Chapter 7, "Uninstalling Integrity Desktop," describes how to remove Integrity Desktop from a computer.
- Chapter 8, "VPN Setup," describes how to configure Integrity Desktop for use on a Virtual Private Network ("VPN").
- Chapter 9, "Automatically Downloading Configuration Files," explains how to cause Integrity Desktop to periodically download a policy file from a central Web server.
- Chapter 10, "Automatically Uploading Alert Logs," explains how to configure Integrity Desktop to periodically upload alert log files to a Microsoft IIS server.
- Appendix B, "Managing Passwords and Files," provides some guidelines for using and storing clear-text passwords and scripts.

## Naming Conventions Used in this Document

There are three Integrity Client operating modes. This document uses the generic term "Integrity Client installer" to refer to three separate installation programs:

- *iclientSetup\_IDen.exe*, which installs Integrity Desktop operating mode
- *iclientSetup\_IFen.exe*, which installs Integrity Flex operating mode
- *iclientSetup\_IAen.exe*, which installs Integrity Agent operating mode

The en in the file name denotes the English language version of Integrity Client.

After installation, this document uses the generic term "Integrity Client program" or "Integrity Client" to refer to settings or operations that apply to all three Integrity Client operating modes. The following table lists the primary differences between the three Integrity Client operating modes.

Default Operating Mode	Works with Integrity Server?	Complete Control Center?	Displays Policies Panel?	Performs Policy Arbitration?
Integrity Desktop	No	Yes	No	No
Integrity Flex	Yes	Yes	Yes	Yes
Integrity Agent	Yes	No	Yes	Yes

# Syntactic Conventions Used in this Document

This document uses the command syntax conventions specified by the Microsoft Manual of Style<sup>1</sup>. The following example illustrates the general form of this syntax:

**SampleParameter**= VariableName { Yes | No } *arguments*...[*options*]

The following table lists the specific elements and syntactic conventions used in this document.

Element	Name	Description
<b>Parameter</b>	Bold	Identifies a specific parameter or command.
{ }	Braces	Indicates a set of choices from which the user must choose.
	OR choice.	Unlike a logical OR, in a configuration parameter statement the pipe symbol separates two mutually exclusive choices. When used in this context, the user types one of the choices, not the symbol.
<i>arguments</i>	Italic	Specifies a variable name or other information the user must provide, such as a path and file name.
...	Ellipsis	Indicates that multiple arguments are repeated in a parameter statement. The user types only the information, not the ellipsis (...).
[ <i>options</i> ]	Brackets	In configuration file parameter statements, brackets indicate optional items. When used to list options, brackets indicates that the user types only the information within the brackets, not the brackets.
[SectionName]	Section Heading	Brackets also identify configuration file section headers. When used to identify the beginning of a configuration file section the brackets must be included.

## Line Breaks and Hyphenation

In a configuration file, an Integrity Client configuration file parameter statement consists of a single line of text. In this document, some example parameter statements may be too long to fit onto a single line. When examples are too long to fit on a single line, line breaks are added; line breaks may also be added to improve the readability of long parameter statements.

Whenever line breaks are added, any additional lines are indented. The following example illustrates the general form of a long configuration file parameter statement to which line breaks and indentation have been added.

```
[Programs]
Default=Allow Local Connect {Allow | Disallow | Ask},
      AllowInternetConnect {Allow | Disallow | Ask},
      AllowLocalServer {Allow | Disallow | Ask},
```

1. Microsoft. Microsoft Manual of Style, 2nd Edition. Microsoft Press, May 1998. ISBN 1-57231-890-2

AllowInternetServer {*Allow* | *Disallow* | *Ask*},  
AllowPassLock {*Yes* | *No*}, Changes Frequently {*Yes* | *No*}

- Examples that have added line breaks are also preceded by the statement “(line breaks added for readability)”.
- Hyphens are never inserted into configuration file parameter statements: if a hyphen appears it must be included as part of the parameter statement.

## Single-step Procedures

In addition to multi-step numbered procedures, this document also contains single-step procedures. The following illustrates the general form of a single-step procedure.

### To perform a single-step procedure:

- Perform the single action described in the body of the single-step procedure.

Most single-step procedures are followed by a brief description of illustration of the results of the single-step procedure’s action.

## Specifying Configuration File or Policy File Parameters

To change the default value for that parameter, type one of the following boolean values:

- Yes, 1, Allow, On, and True are equivalent values
- No, 0, Disallow, Off, and False are equivalent values

## Notes and Cautions

This document two types of specially annotated text: notes and cautions.

## Notes



Notes emphasize related, reinforcing, interesting, or other “special” information.

## Cautions



Cautions identify actions or processes that can potentially damage data or programs.

## Single-Step Procedures

In addition to multi-step numbered procedures, this document also contains single-step procedures. The following illustrates the general form of a single-step procedure.

### **To perform a single-step procedure:**

➤ Perform the single action described in the body of the single-step procedure.

Most single-step procedures are followed by a brief description of illustration of the results of the single-step procedure’s action.

# Chapter 2

---

## System Requirements

The following sections list the hardware and software components required to successfully install and run the Integrity Desktop version 4.0 program.

### Recommended System Requirements

Integrity Desktop is optimized for use under the recommended requirements listed below:

- Microsoft® Windows® 95 OSR2, 98 SE, NT 5, SP6a, 2000 Professional SP 3 or greater, XP (all SPs)
- IBM PC or 100% compatible
- Pentium II processor 450 MHz or higher
- 128 MB or higher RAM
- 10 MB Hard disk space

All versions of Windows must include Internet Explorer 5.0 or greater.

### Minimum System Requirements

Integrity Desktop is functional but not optimized under the minimum requirements listed below.

- Microsoft® Windows® 95 OSR2, 98 SE, NT 5, SP6a, 2000 SP 3 or greater, XP (all SPs)
- IBM PC or 100% compatible
- Pentium processor 233 MHz or higher
- 32 MB RAM
- 10 MB Hard disk space

All versions of Windows must include Internet Explorer 5.0 or greater.

### Windows System Requirements

Microsoft Windows® has minimum system resource requirements. Contact Microsoft® or visit [www.microsoft.com](http://www.microsoft.com) for more information.

## Upgrading to Integrity Desktop 4.0

Zone Labs, Inc. designed Integrity Desktop 4.0 for easy upgrade. However, some older Zone Labs products, particularly products that were not designed for use in an enterprise setting, have special upgrade requirements.

### Upgrade Guidelines

The following table lists the upgrade paths from older Zone Labs products to Integrity Desktop 3.7

	<b>To</b>						
<b>From</b>	Zone Alarm Pro - Integrity Desktop 3.1	Integrity Agent 1.0 <i>(no longer supported)</i>	Integrity Agent 1.5 <i>(no longer supported)</i>	Integrity Client 3.5	Integrity Client 3.7	Integrity Client 4.0	
Zone Alarm Pro - Integrity Desktop 3.1	—	D	D	U	U	U	
Integrity Agent 1.0 <i>(no longer supported)</i>	U	—	U	U	U	U	
Integrity Agent 1.5 <i>(no longer supported)</i>	U	D	—	U	U	U	
Integrity Client 3.5	D	D	D	—	U	U	
Integrity Client 3.7	D	D	D	D	—	U	
Integrity Client 4.0	D	D	D	D	D	—	
D = Downgrade after uninstalling previous version U = Exclusive Upgrade							

The specific cases listed in the preceding table produce the following general rules governing upgrade:

- All uninstalls remove all existing user settings. The only way to save user settings is to perform an upgrade.

- Installation of consumer products (such as Zone Alarm or Zone Alarm Pro) over Integrity enterprise products is not permitted: You must uninstall the enterprise-level product before installing the consumer product.
- Downgrading deletes all existing user settings.
- Downgrading the release level of a product is not allowed: you must uninstall the higher-numbered release before installing the lower numbered release.
- Upgrades preserve user settings, but allows the user the choice of removing existing settings.

## Upgrading, SMS, and Rebooting

To complete an upgrade of an existing instance of a Zone Labs product to Integrity Client, it is necessary to reboot the computer to complete the upgrade process.

To allow the deployment of Integrity Client with third-party systems management tools such as Microsoft System Management Server (SMS), use the `/noreboot` installation command line switch to defer reboot to a more convenient time. The key word is *defer*: the computer must eventually be rebooted to complete the upgrade process.

Reboot of the computer, and thus the use of the `noreboot` switch, is not required during an initial (sometimes referred to as “clean”) installation of Integrity Client. Instead, to properly initialize Windows settings and variables a newly installed Integrity Client must be run for the first time while the computer has an administrator-level user logged in.

## Upgrading and Policy Deployment

After upgrading, updated instances of Integrity Client receive fresh enterprise security policies from Integrity Server.

The following sections describe three specific upgrade scenarios.

## Specific Upgrade Scenarios

This section describes specific issues for three specific upgrade scenarios:

- “Upgrading ZoneAlarm Pro 2.6 to Integrity Desktop 4.0,” in the following section
- “Upgrading Zone Alarm Pro - Integrity Desktop 3.1 to Integrity Desktop 4.0,” on page 11
- “Upgrading ZoneAlarm Pro Users to Integrity Desktop 4.0,” on page 12

This chapter also includes a brief review of the Integrity Desktop installation-level passwords.

## Upgrading ZoneAlarm Pro 2.6 to Integrity Desktop 4.0

Choose one of the following procedures based on whether or not a ZoneAlarm Pro 2.6 installation-level password has been set.

## If the ZoneAlarm Pro 2.6 Installation-level Password is Not Set

Complete the following tasks when a ZoneAlarm Pro 2.6 password installation has not been set:

- 1 Run the ZoneAlarm Pro 2.6 uninstall program.
- 2 Manually remove the ZoneAlarm Pro 2.6 database files.
- 3 Install Integrity Desktop and set an Integrity Desktop installation-level password.

## If the ZoneAlarm Pro 2.6 Installation-level Password is Set

Complete the following major steps when a ZoneAlarm Pro 2.6 password installation is set:

- 1 Run the ZoneAlarm Pro 2.6 uninstall program.
- 2 Decide whether or not to preserve existing ZoneAlarm Pro 2.6 user security preferences.
- 3 Install Integrity Desktop and supply the ZoneAlarm Pro installation-level password.
  - To preserve existing user security preferences, do not include the `/reset` switch in the installation command line.
  - To remove existing user security preferences, include the `reset` switch in the installation command line. See "Using the Reset Command Line Parameter," in the following section, and `/reset`, on page 24, for more information about the `reset` switch.

## Using the Reset Command Line Parameter

During installation, Integrity Client attempts to use the settings contained in the ZoneAlarm Pro 2.6 database.

The use of the `reset` parameter, described on page 24, ensures that existing ZoneAlarm Pro 2.6 personal policy's settings will be removed. Otherwise, Integrity Desktop's personal policy can only be modified by importing settings from a configuration file (default name `policy.ini`) on the local machine.

## Upgrading Zone Alarm Pro - Integrity Desktop 3.1 to Integrity Desktop 4.0

Complete the following major steps to upgrade Zone Alarm Pro - Integrity Desktop 3.1 to Integrity Desktop 4.0:

- 1 Decide whether or not to preserve existing Zone Alarm Pro - Integrity Desktop 3.1 user security preferences.
- 2 Install Integrity Desktop 4.0 over Zone Alarm Pro - Integrity Desktop 3.1 and supply the installation-level password.
  - To preserve existing user security preferences, do not include the `reset` switch in the installation command line.

- To remove existing user security preferences, include the reset switch in the installation command line. See ““Using the Reset Command Line Parameter,” on page 11,” and “Reset,” on page 24, for more information about the reset switch.

## Upgrading ZoneAlarm Pro Users to Integrity Desktop 4.0

Any time a consumer-level product is upgraded to Integrity Desktop, the consumer product's security and preferences settings are lost.

---

## Installing Integrity Desktop 4.0

---

This chapter describes how to install Integrity Desktop 4.0.

This chapter is organized into the following main sections:

- “Performing a Default Installation,” in the following section, describes how to perform a default installation of Integrity Desktop
- “Performing a Non-default Installation,” on page 14, describes how to use installation command line switches to change the default behaviors of the Integrity Desktop installation program
- “Post-installation Configuration Files,” on page 34, provides an overview of the use of policy files to configure Integrity Desktop’s initial settings after installation has been completed.

### Before You Begin

To complete the procedure in this chapter, you will need a copy of the Integrity Desktop installation program *iclientSetup\_IDen.exe*.

### Initial Installation, SMS, and Reboot

Unlike upgrades, initial (sometimes referred to as “clean”) installations of Integrity Client do not require the reboot of the upgraded computer.

Instead, to properly initialize Windows settings and variables a newly installed Integrity Client must be run for the first time while the computer has an administrator-level user logged in.

### Performing a Default Installation

Unlike Integrity Flex or Integrity Agent, Integrity Desktop does not operate in conjunction with a corresponding Integrity Server. This means that the configuration, deployment, installation, and administration tools provided by Integrity Server are not available.

Complete the following single-step procedure to perform a default installation of Integrity Desktop.

#### To perform a default installation of Integrity Desktop:



- Double-click the Integrity Desktop installation program *iclientSetup\_IDen.exe*.

The installation program starts. As part of the default installation process, a series of dialog boxes and information displays appears.

## Default Installation

Run in its default state, during an initial installation the Integrity Desktop installation program:

- Prompts for installation options and user information
- Asks whether or not to run the setup wizard
- Asks whether or not to run the Integrity Desktop tutorial
- Asks whether or not to start the Integrity Desktop program

The next section describes how to use installation command line switches to specify a non-default installation.

## Performing a Non-default Installation

Use installation command lines to change the default installation and first-time startup behaviors of the Integrity Desktop installation program.

## Installation Command Lines

There are two distinct types of Integrity Client command lines

- Installation command lines, described in this chapter
- Operational command lines described in “Operational Command Lines,” on page 36

The following table illustrates the primary differences between the two types of command lines.

Operational Characteristic	Installation Command Line	Operational Command Line
When used	During installation	After installation
Used with file	Integrity Client Installation program <i>iclientSetup_IXen.exe</i> . <sup>a</sup>	Integrity Client program file <i>iclient.exe</i> .
Parameter delimiter	Slash mark (“/”)	Dash (“-”)
Configuration file specifier	<ul style="list-style-type: none"> <li>• Does not include a special preceding command line switch</li> <li>• Path and file name specifier must be enclosed in quotation marks (“”)</li> <li>• Must be the last switch on an installation command line</li> </ul>	<ul style="list-style-type: none"> <li>• Must be preceded by the <code>-config</code> command line switch</li> <li>• Path and file name specifier must be enclosed in quotation marks (“”)</li> <li>• Must be the last switch on an operational command line</li> </ul>

a. Where *IX* equals *ID* for Integrity Desktop, *IF* for Integrity Flex, of *IA* for Integrity Agent, and *en* is the language code.

## Overview of Installation Command Lines

The following illustrates the general form of an Integrity Client installation command line (line break added for readability):

```
iclientSetup_110n.exe
[/switch_1 /switch_2 ... /switch_n] ["C:\full\path\to\configuration.ini"]
```

The installation command line consists of three primary elements:

- `iclientSetup_110n.exe` is the name of the Integrity Client installation program, where *n* is 1, 2, or 3, depending on client type.
- Optional command line switches, preceded by the slash mark ("/"), specify non-default installation and post-installation behaviors.
- `C:\full\path\to\configuration.ini` specifies the path to an optional installation configuration file to be loaded by Integrity Client after installation is completed.

## Limitations on Installation Command Line Length

Different versions of Microsoft Windows place differing constraints on the maximum size of installation command lines.

The following table contains the known limitations for installation command lines supplied directly to different versions of Microsoft Windows, as well as for installation command lines included as part of an Integrity Server installation package.

Windows Version	Maximum Installation Command Line Length (characters + spaces)
Command line installation values	
98 SE	127
NT, 2000, XP	277
Integrity Server client deployment package values	
98	219
NT	226
2000	195
XP	199

## The Configuration File Installation Command Line Specifier

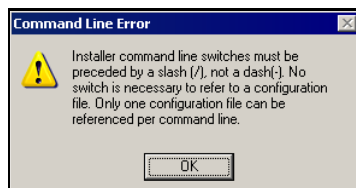
Special syntactic rules apply to the installation configuration file command line specifier ("*C:\full\path\to\configuration.ini*" in the example in the preceding section). If specified in an installation command line, the configuration file specifier:

- Must be the last element on the command line
- Must *not* be prefaced by a slash. This is the only command line element that does not require a delimiter character.
- Must enclose the path name and filename in quotation marks ("")
- Can use Microsoft Windows' Universal Naming Convention (UNC) of `\\servername\sharename` to refer to a policy file located on a shared network resource

When the installation configuration file command line specifier is used, Integrity Client ignores the `Policy_Info` section of the specified configuration file.

## Installation Command Line Error Messages

If you use a dash delimiter ("-") in an installation command line, the Integrity Client installation programs displays the following error message.



If you use a dash delimiter ("-") in an installation command line, the Integrity Client installation program displays this Command Line Error message box.

## Installation Command Line Switches

All installation command line switches are preceded by a slash mark ("/").

Integrity Client recognizes eighteen installation command line switches (seventeen for Integrity Desktop). The following table groups the installation command line switches into four functional categories and identifies the page in this chapter where a complete description of the switch can be found.

Command Line Switch	Description	Page
General Installation Command Line Switches		
<code>/errlog Path</code>	Specifies an installation error log file.	18
<code>/forceupgrade</code>	Suppress the display of the <i>Previous Settings</i> dialog box.	18
<code>/install_log Path</code>	Specifies a non-default location for the installation log file.	19
<code>/installdir Path</code>	Specifies a non-default location for Integrity Client program files.	19
<code>/lickey LicenseKey</code>	Specifies the product license key.	20
<code>/noreboot</code>	Suppresses automatic rebooting after an upgrade.	20
<code>/nostartup</code>	Suppresses automatic startup of Integrity Client at boot.	21


Command Line Switch	Description	Page
<code>/notminimized</code>	After installation, display the Integrity Client Control Center.	21
<code>/rbprompt</code>	Overrides silent install by displaying a reboot prompt.	22
<code>/reboot</code>	Force a reboot after installation.	22
<code>/regfile</code>	Specifies the path to a file containing Windows Registry entries.	22
<code>/reset</code>	Clears existing Zone Labs configuration settings.	23
<code>/s</code>	Specifies silent (prompt-free) installation.	24
<code>/upgradekey</code>	Supplies an existing upgrade key.	26
<code>/upgradekeyset</code>	Specifies a new upgrade key.	24
<b>Tutorial and Wizard Installation Command Line Switches</b>		
<code>/notutorial</code>	Suppresses display of the product tutorial.	27
<code>/nowizards</code>	Suppresses display of the configuration wizard.	28
<code>/i</code>	Suppresses both the product tutorial and configuration wizard.	28
<b>Set or Modify Password Command Line Switches</b>		
<code>/passwd <i>UserPwordNew</i></code>	Specifies a new optional user-level password.	29
<code>/password <i>UserPwordOld</i></code>	Supplies an existing user-level password.	29
<code>/pwinstset <i>InstallPwordNew</i></code>	Specifies a new optional installation-level password.	31
<code>/pwinst <i>InstallPwordOld</i></code>	Supplies an existing installation-level password.	32
<b>Specify an optional installation configuration file</b>		
<code>"<i>Path to Configuration File</i>"</code>	Specifies the path and name of an optional installation configuration file.	33
<b>For networks with Integrity Server only, specify an optional installation policy file</b>		
<code>/policy "<i>Path to Policy File</i>"</code>	Specifies the path and name of an optional installation policy file.	33

## General Installation Command Line Switches

Use the General installation command line switches group to specify:

- Non-default installation behaviors
- Non-default locations for the post-installation folders and files used by Integrity Client

The following tables list the nine general installation command line switches in alphabetical order.

General Installation Command Line Switches	
<b>/errlog</b> <i>Path</i>	
<p>Use <code>errlog</code> to specify an error log file's name and storage location.</p> <p>The following illustrates the general form of the <code>errlog</code> installation command line switch (line break added for readability):</p> <pre>IDSetup_1101.exe /errlog "C:\PathName\ErrorLogFileName.txt" ... "C:\Path\To\Configuration.ini"</pre> <p>The path specifier:</p> <ul style="list-style-type: none"> <li>• Must be enclosed in quotation marks ("")</li> <li>• Can use Microsoft Windows' Universal Naming Convention (UNC) of <code>\servername\sharename</code> to refer to an installation configuration file located on a shared network resource</li> </ul>	
	<p>If <code>errlog</code> is used in a command line with the <code>/s</code> ("silent") switch, described on page 24, the <code>s</code> switch must immediately precede the <code>errlog</code> command.</p>
<p>The following illustrates the use of the <code>errlog</code> installation command line switch in conjunction with the <code>s</code> installation command line switch (line break added for readability):</p> <pre>IDSetup_1101.exe [/s] /errlog "C:\PathName\ErrorLogFileName.txt" /... C:\Path\to\ErrorLog.txt"</pre> <p>Specifying the <code>s</code> switch without the <code>errlog</code> switch automatically creates an error log file named <code>ErrorLog.txt</code> and saves it in the Integrity Client program folder at <code>C:\Program Files\Zone Labs\Integrity Client\</code>. To modify the default behavior of the <code>s</code> switch, use the <code>errlog</code> switch to specify a different path and file name. See the <code>s</code> switch for more information.</p> <p>Default Value: None—<code>ErrLog</code> must include a path and file name specifier.</p>	

General Installation Command Line Switches	
<b>/forceupgrade</b>	
<p>Use <code>forceupgrade</code> to suppress the <b>Previous Settings</b> dialog box that offers the user the choice of overwriting their existing settings during the upgrade process: This has the effect of forcing users to retain their existing Integrity Client settings.</p> <p>The following illustrates the general form of the <code>forceupgrade</code> installation command line parameter:</p> <pre>iclientSetup_1101.exe /forceupgrade</pre> <p>When used on the same installation command line as the <code>/s</code> switch, the <code>forceupgrade</code> switch has no effect.</p> <p>Default: No default value.</p>	


General Installation Command Line Switches	
<b>/install_log Path</b>	
Use <code>install_log</code> to specify a secure non-default destination for the installation log file <code>Install.Log</code> .	
The following illustrates the general form of the <code>install_log</code> installation command line switch:	
<code>iclientSetup_1101.exe /install_log "C:\Full\Path\To\InstallLog.txt"</code>	
<ul style="list-style-type: none"> <li>• The folder specified by the path name must be created before specifying <code>install_log</code>: The <code>install_log</code> installation command line switch can <i>not</i> be used to create a new folder.</li> <li>• When using <code>install_log</code>, always enclose the complete path name in quotation marks (").</li> </ul>	
If <code>install_log</code> was used to place the install log in a non-default location, specify the location of the install log at uninstallation as follows:	
<code>zauninst.exe "c:\PathToInstallLog\InstallLog.txt"</code>	
Default Value: "C:\Program Files\Zone Labs\Integrity Client\". Zone Labs recommends that the default folder name be used.	

General Installation Command Line Switches	
<b>/installdir Path</b>	
Use <code>installdir</code> to specify an alternative destination for the Integrity Client program files. The following illustrates the general form of the <code>installdir</code> installation command line switch:	
<code>iclientSetup_1101.exe /installdir "C:\Program Files\Folder"</code>	
<ul style="list-style-type: none"> <li>• The <code>installdir</code> switch specifies where Integrity Client program files are stored: <code>installdir</code> does not change the storage locations of Integrity Client database files.</li> <li>• When using <code>installdir</code>, always enclose the complete path name in quotation marks (").</li> <li>• Do not use <code>installdir</code> and the <code>/s</code> switch, described on page 24, in the same installation command line: if <code>installdir</code> and the <code>s</code> switch, described on page 24, are used in the same command line, Integrity Client can not display errors resulting from invalid path and filename specifications.</li> </ul>	
Default Value: C:\Program Files\Zone Labs\Integrity Client\. Zone Labs, Inc. recommends that the default folder name be used.	




General Installation Command Line Switches
<b>/nostartup</b>
<p>Use <code>nostartup</code> to specify that the Integrity Client installation program not ask whether to start the program after an initial installation.</p> <p>The following illustrates the general form of the <code>nostartup</code> installation command line switch:</p> <pre>iclientSetup_1101.exe /nostartup</pre> <p>Because the <code>nostartup</code> installation command line switch does not provide the user with an opportunity to respond to the startup prompt, the newly installed instance of Integrity Client will not be started after installation.</p> <p>Default Value: Off. Unless specified by <code>nostartup</code>, the installation program asks to start Integrity Client after an initial installation.</p>


General Installation Command Line Switches
<b>/notminimized</b>
<p>Use <code>notminimized</code> to force the display of the Integrity Client Control Center when Integrity Client starts for the first time after installation.</p> <p>When the <code>/s</code> switch is included as part of an installation command line, the Integrity Client installation program starts Integrity Client for the first time in so-called “minimized” mode: Only the Integrity icon appears in the Windows system tray. The <code>notminimized</code> installation command line switch overrides this default behavior.</p> <p>Default Value: Off (Control Center is minimized) for installations that include the <code>/s</code> installation command line switch.</p>

General Installation Command Line Switches	
<b>/rbprompt</b>	
<p>Use <code>rbprompt</code> in conjunction with the <code>s</code> (“silent”) switch, described on page 24, to prompt the user to perform the reboot required to complete an upgrade of Integrity Client; the reboot prompt is only displayed if reboot is required by the upgrade process.</p> <p>The following illustrates the general form of the <code>rbprompt</code> installation command line switch:</p> <pre>iclientSetup_1101.exe /s /rbprompt</pre>	
	<p><b>The <code>rbprompt</code> can only be used in conjunction the <code>s</code> switch: <code>rbprompt</code> allows a reboot prompt, and only a reboot prompt, to be displayed as part of a silent upgrade.</b></p>
<ul style="list-style-type: none"> <li>• If <code>rbprompt</code> is specified as part of an upgrade of Integrity Client that is managed by a third-party installer setup tool such as Microsoft’s SMS, <code>rbprompt</code> will require a response to the reboot prompt before allowing the installer setup script to continue.</li> <li>• Integrity Server’s Client Deployment feature automatically includes the “<code>/s /rbprompt</code>” command pair as part of an Integrity Client installation package. To reboot automatically after an upgrade do not select the <b>Run installer without UI...</b> check box. Instead, in the <b>Additional Commands</b> text entry area, specify the <code>s</code> command line switch without a corresponding <code>/rbprompt</code> switch.</li> <li>• Using <code>rbprompt</code> on the same installation command line as the <code>noreboot</code> installation command line switch, described on page 20, suppresses the display of the reboot prompt dialog box: <code>noreboot</code> defers the reboot to the controlling third-party installation setup tool, such as SMS. (As described in the description of <code>/noreboot</code>, an upgrade is not complete until a reboot has been performed).</li> </ul> <p>Default Value: Use <code>rbprompt</code> to modify the default operation of the <code>s</code> switch. Unless explicitly specified by <code>rbprompt</code>, the <code>s</code> switch suppresses all messages, and after an upgrade (as distinguished from a clean install) automatically reboots the computer.</p>	

General Installation Command Line Switches	
<b>/reboot</b>	
<p>Use <code>reboot</code> to force a reboot of Integrity Client after installation.</p> <p>Normally, when the Integrity Client installation program does not detect files from an existing Zone Labs product during the installation process, the computer is not automatically rebooted. Use the <code>reboot</code> switch to force a reboot under all circumstances.</p> <p>Default: No default value.</p>	

General Installation Command Line Switches	
<b>/regfile</b>	<p>Use the <code>regfile</code> switch to have the Integrity Client installation program apply Windows Registry keys and values contained in a “.reg” file to the Windows Registry at the time of installation.</p> <p>The following illustrates the general form of the <code>regfile</code> command.</p> <pre>iclientSetup_1101.exe /regfile="c:\full\path\to\registry\RegFile.reg"</pre> <p>Any valid Windows filename can be used, but the <code>.reg</code> file must:</p> <ul style="list-style-type: none"> <li>• Contain valid Windows Registry keys and values</li> <li>• Use the <code>.reg</code> file name extension</li> </ul> <p>When creating a client installation package with Integrity Server, you can include a <code>.reg</code> file in an installation package. The <code>/regfile</code> switch directs the Integrity Client installation program to apply the keys and values of the <code>.reg</code> file to the Windows Registry.</p> <p><b>To include a registry file in the client installation package:</b></p> <ol style="list-style-type: none"> <li>1 Create a package using the <b>Client Deployment   New Package</b> screen.</li> <li>2 In the Integrity Server folder hierarchy, navigate to the folder containing the package you just created. The following illustrates the default path (line break added):.0 <pre>c:\Program Files\ZoneLabs\Integrity\jakarta-tomcat-n.n.n\ webapps/integrity/package/PackageName</pre> </li> <li>3 In the folder specified by <code>PackageName</code>: <ol style="list-style-type: none"> <li>a Create a new folder named <code>extras</code>.</li> <li>b Place the <code>.reg</code> file in the <code>extras</code> folder.</li> </ol> </li> <li>4 In Integrity Server, return to the <b>Client Deployment   List</b> dialog box, select the installation package, and click <b>Edit</b>. <p>The Client Deployment’s <b>Edit Package</b> screen appears.</p> </li> <li>5 In the <b>Install Parameters</b> section, in the <b>Additional Command Line Switches</b> text entry area, add the command line switch <code>/regfile</code>.</li> <li>6 Click <b>Save</b>.</li> </ol> <p>A registry file can also be referenced by the Policy Update Utility.</p>

General Installation Command Line Switches	
<b>/reset</b>	
Use <code>reset</code> during upgrade or reinstallation to completely clear all Integrity Client settings. The following illustrates the general form of the reset installation command line switch:  <code>iclientSetup_1101.exe /pwinst <i>InstallPasswordOld</i> /reset</code>	
If an installation-level password was specified during initial installation, the <code>pwinst</code> switch must appear on the same command line with <code>reset</code> .	
Default Value: Off.	
	The <code>reset</code> installation command line switch must be used with caution. After using <code>reset</code> , all Integrity Client personal policy settings except the installation-level password are lost and must be reinitialized.

General Installation Command Line Switches	
<b>/s</b>	
Use <code>s</code> (for “silent”) to suppress all Integrity Client installation program messages.	
	<b>If used, the <code>s</code> switch must be the first switch on the installation command line.</b>

General Installation Command Line Switches <i>(continued)</i>
<b>/s</b>
<p>The following illustrates the general form of the <code>s</code> installation command line switch:</p> <pre>iclientSetup_1101.exe /s</pre> <p>If used, the <code>s</code> switch:</p> <ul style="list-style-type: none"><li>• Must be the first switch on the installation command line.</li><li>• Forces a reboot if the installer detects files from an existing Zone Labs product on the computer, and those files cannot be replaced at the time the installation or upgrade of Integrity Client is performed. This is true even if the <b>Clean Install</b> check box is selected by the user.</li><li>• Automatically creates an error log file named <code>ErrorLog.txt</code> and saves it in the Integrity Client program folder. To change the default path and file name of the Integrity Client program folder, use the <code>errlog</code> switch.</li></ul> <p>Do not use <code>installdir</code> and the <code>/s</code> switch in the same installation command line. If <code>installdir</code> and <code>s</code> are used together on the same command line, errors resulting from invalid path and filename specifications will not be displayed during installation.</p> <p>Integrity Client does not allow the TrueVector security engine to be shut down silently unless an installation-level password is supplied.</p> <p>There are two conditions that affect how an upgrade will or will not be performed:</p> <ul style="list-style-type: none"><li>• An installation-level password was set for the existing installation, and you supply the installation-level password on the command line during re-installation, then a silent installation is performed. If the installation-level password is not correctly specified, the upgrade fails silently.</li><li>• An upgrade key was set for the existing installation, and you supply the upgrade key on the command line during re-installation, then a silent installation is performed. If the upgrade key is not correctly specified, the upgrade is performed but not silently.</li></ul> <p>The following illustrates the use of the <code>s</code> command line switch in conjunction with the <code>pwinst</code> switch:</p> <pre>iclientSetup_1101.exe /s /pwinst InstallPwordOld</pre> <p>See <b>pwinst</b>, on page 32, for more information.</p> <p>Default value: Off. Unless explicitly disabled by the use of <code>s</code>, messages and prompts are displayed by the Integrity Client installation program.</p>

General Installation Command Line Switches
/upgradekey
<p>Use the <code>upgradekey</code> switch to specify an existing upgrade key. The following illustrates the general form of the <code>upgradekey</code> switch:</p> <pre data-bbox="467 436 1024 464">iclientSetup_1101.exe /upgradekey <i>upgradeKeyOld</i></pre> <ul data-bbox="448 485 1417 695" style="list-style-type: none"><li>• Use the <code>/upgradekeyset</code> installation command line switch, described in the following table in this section, to create a new upgrade key during initial installation.</li><li>• Use the <code>/upgradekey</code> and <code>/upgradekeyset</code> installation command lines on the same command line to change the value of an existing upgrade key during a re-installation.</li><li>• Use the <code>-upgradekey</code> operational command line switch, described on page 1, to specify an existing upgrade key during reconfiguration of an existing instance of Integrity Client.</li></ul> <p>The upgrade key suppresses:</p> <ul data-bbox="448 762 1403 890" style="list-style-type: none"><li>• Any dialogs that normally appear during reconfiguration or upgrade. Contrast this with the installation-level password which prevents anyone from uninstalling or upgrading Integrity Client without supplying the password.</li><li>• The TrueVector shutdown dialog box.</li></ul> <p>For example, if an upgrade key is set, and someone attempts to reconfigure or re-install without supplying the upgrade key, the Integrity Client installation program completes the upgrade: Any upgrade dialogs will, however, be shown.</p> <p>The Integrity Client installation program suppresses dialogs if an installation-level password is specified. This means upgrades performed in conjunction with an installation-level password, the upgrade key does not also need to be specified.</p> <p>Use the <code>upgradekeyset</code> installation command line switch, described in the next table in this section, to specify the upgrade key during initial installation. After initial installation, use the <code>upgradekey</code> operational command line switch, described on page, to change an existing upgrade key.</p> <p>Default: No default value.</p>

General Installation Command Line Switches	
<b>/upgradekeyset</b>	
	<p>Use the <code>upgradekeyset</code> switch to create a new upgrade key at the time Integrity Client is installed. The following illustrates the general form of the upgrade key switch:</p> <pre>iclientSetup_1101.exe /upgradekeyset <i>upgradeKeyNew</i></pre> <ul style="list-style-type: none"> <li>• Use the <code>/upgradekey</code> installation command line switch, described in the previous table in this section, to specify a silent (prompt free) upgrade of an existing installation.</li> <li>• Use the <code>/upgradekey</code> and <code>/upgradekeyset</code> installation command lines on the same command line to change the value of an existing upgrade key during a re-installation.</li> <li>• Use the <code>-upgradekey</code> operational command line switch, described on page 1, to specify an existing upgrade key during reconfiguration of an existing instance of Integrity Client.</li> </ul> <p>The upgrade key suppresses the dialogs that normally appear during reconfiguration or upgrade. Contrast this with the installation-level password which prevents anyone from uninstalling or upgrading Integrity Client without supplying the password.</p> <p>For example, if an upgrade key is set, and someone attempts to reconfigure or re-install without supplying the upgrade key, the Integrity Client installation program completes the upgrade: Any upgrade dialogs will, however, be shown.</p> <p>The Integrity Client installation program suppresses dialogs if an installation-level password is specified. This means upgrades performed in conjunction with an installation-level password, the upgrade key does not also need to be specified.</p> <p>Default: No default value.</p>

## Tutorial and Wizard Installation Command Line Switches

Use the tutorial and wizard command line switches group to specify whether or not the Integrity Client tutorial and wizard are displayed as part of the installation process. The following tables list the three tutorial and wizard command line switches.

Tutorial and Wizard Installation Command Line Switches	
<b>/notutorial</b>	
	<p>Use <code>notutorial</code> to suppress the automatic display of the Integrity Client tutorial after installation is completed. The following illustrates the general form of the <code>notutorial</code> installation command line switch:</p> <pre>iclientSetup_1101.exe <b>/notutorial</b></pre> <p>Default Value: Off. If not explicitly disabled by the use of <code>notutorial</code>, the installation program asks the user if they want to view the tutorial as part of an initial installation.</p>

Tutorial and Wizard Installation Command Line Switches	
<b>/nowizards</b>	
Use <code>nowizards</code> to suppress the automatic display of the Integrity Client configuration wizard after installation is completed. The following illustrates the general form of the <code>nowizards</code> command line switch:	
<code>iclientSetup_1101.exe /nowizards</code>	
Default value: Off. If not explicitly disabled by the use of <code>nowizards</code> , the installation program asks if the user wants to run the configuration wizard as part of an initial installation.	

Tutorial and Wizard Installation Command Line Switches	
<b>/i</b>	
Use <code>i</code> to combine the operation of both the <code>notutorial</code> and <code>nowizards</code> command line switches. The following illustrates the general form of the <code>i</code> installation command line switch:	
<code>iclientSetup_1101.exe /i</code>	
In this example, the <code>i</code> switch suppresses both the automatic start of the Integrity Client tutorial and the automatic start of the Integrity Client configuration wizard after installation is completed.	
Default value: Off.	

## Set or Modify Password Installation Command Line Switches

Integrity Desktop recognizes both a user-level and an installation-level password.



**Zone Labs, Inc. recommends you *not* set a user-level password. A user-level password prevents the end-user from responding to Integrity Desktop alerts and interferes with the application of centrally administered updates and changes.**


The following table lists the functional differences between the two password types.

Function	User-level Password	Installation-level Password
Enable override of user-level password		✓
Enable silent installations, uninstalls, or upgrades		✓
Prevent changes to personal security settings	✓	
Prevent shutting down Integrity Desktop	✓	
Prevent uninstalling Integrity Desktop	✓	✓
Settable from Control Center	✓	
Settable from installation command line ("/" delimiter)	✓	✓
Changeable from operational command line ("-." delimiter)	✓	✓

Use the set or modify password installation command line switches group to:

- Set passwords during installation
- Change existing passwords during reinstallation
- Enable changes to an existing instance of Integrity Client

The following tables list the four set or modify passwords command line switches.

Set or Modify Password Installation Command Line Switches	
<b>/passwset</b> <i>UserPwordNew</i>	
	<p>Use <code>passwset</code> to define a new user-level password.</p> <p>A user-level password:</p> <ul style="list-style-type: none"> <li>• Must be a minimum of 6 characters and a maximum of 31 characters, and cannot contain spaces</li> <li>• Can only be set when no Integrity Client database files (“<code>.rdb</code>” file name extension) are present in the computer’s <code>C:\%windir%\Internet Logs</code> folder</li> </ul> <p>The following illustrates the general form of the <code>passwset</code> installation command line switch:</p> <pre>iclientSetup_1101.exe /passwset UserPwordNew</pre> <p>Zone Labs, Inc. recommends that a user-level password <i>not</i> be set during initial installation of Integrity Client. A user-level password prevents the end-user from responding to Integrity Client alerts and interferes with the application of centrally administered updates and changes.</p> <p>Default Value: No default value.</p>

Set or Modify Password Installation Command Line Switches	
<b>/password</b> <i>UserPwordOld</i>	
	<p>Use the <code>password</code> switch to supply a previously defined user-level password to the Integrity Client installation program. The following illustrates the general form of the <code>password</code> installation command line switch:</p> <pre>iclientSetup_1101.exe /password UserPwordOld</pre> <p>After installation, the <code>password</code> switch can be used in conjunction with <code>passwset</code> (described in the preceding table entry) to update an existing user-level password. In the following, <code>password</code> enables an existing user-level password to be modified:</p> <pre>iclientSetup_1101.exe /password UserPwordOld /passwset UserPwordNew</pre> <p>Default Value: Not applicable during initial installation.</p>

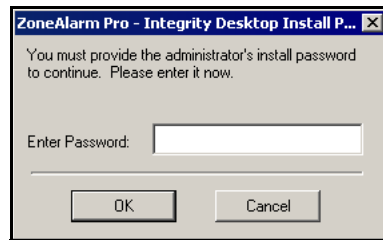


## Set or Modify Password Installation Command Line Switches

### `/pwinstset InstallPwordNew`

Use `pwinstset` to define a new installation-level password. An installation-level password prevents unauthorized changes to an existing Integrity Client installation.

If an installation-level password was set during installation, and a user attempts to uninstall Integrity Client without specifying the installation-level password, the following dialog box appears.



#### **Install Password dialog box.**

If the correct installation level password is not supplied, the uninstallation process stops.

- An installation-level password must be a minimum of 6 characters and a maximum of 31 characters, and can not contain spaces.

Installation-level passwords do not affect the user's ability to change his or her personal security settings.

Installation-level passwords can be:

- *Set* from the command line only during initial installation
- *Changed* during reinstallation if the `pwinst` switch appears on the same installation command line to enable the change

The `reset` switch, does *not* clear the installation password.

Integrity Client provides no other methods for changing or updating an installation-level password.

Set or Modify Password Installation Command Line Switches <i>(continued)</i>						
<p><b>/pwinstset <i>InstallPwordNew</i></b></p> <p>The following table inset illustrates three uses of the pwinstset installation command line switch.</p> <table border="1"> <thead> <tr> <th>Initial installation</th> </tr> </thead> <tbody> <tr> <td> <pre>iclientSetup_1101.exe /pwinstset <i>InstallPwordNew</i></pre> <ul style="list-style-type: none"> <li>• In this example pwinstset sets the installation-level password for the first time.</li> </ul> </td> </tr> <tr> <th>Changing an installation-level password without the reset switch.</th> </tr> <tr> <td> <pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNew</i></pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• Pwinstset changes the installation-level password</li> </ul> </td> </tr> <tr> <th>Clearing the user-level password with the reset switch (line break added).</th> </tr> <tr> <td> <pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNe</i> /reset</pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable specifying a new installation-password</li> <li>• Pwinstset specifies a new installation-level password</li> <li>• Reset clears the existing user-level password</li> </ul> </td> </tr> </tbody> </table> <p>Default Value: No default value.</p>	Initial installation	<pre>iclientSetup_1101.exe /pwinstset <i>InstallPwordNew</i></pre> <ul style="list-style-type: none"> <li>• In this example pwinstset sets the installation-level password for the first time.</li> </ul>	Changing an installation-level password without the reset switch.	<pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNew</i></pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• Pwinstset changes the installation-level password</li> </ul>	Clearing the user-level password with the reset switch (line break added).	<pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNe</i> /reset</pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable specifying a new installation-password</li> <li>• Pwinstset specifies a new installation-level password</li> <li>• Reset clears the existing user-level password</li> </ul>
Initial installation						
<pre>iclientSetup_1101.exe /pwinstset <i>InstallPwordNew</i></pre> <ul style="list-style-type: none"> <li>• In this example pwinstset sets the installation-level password for the first time.</li> </ul>						
Changing an installation-level password without the reset switch.						
<pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNew</i></pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• Pwinstset changes the installation-level password</li> </ul>						
Clearing the user-level password with the reset switch (line break added).						
<pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNe</i> /reset</pre> <p>In this example:</p> <ul style="list-style-type: none"> <li>• Pwinst specifies the existing installation-level password to enable specifying a new installation-password</li> <li>• Pwinstset specifies a new installation-level password</li> <li>• Reset clears the existing user-level password</li> </ul>						

Set or Modify Password Installation Command Line Switches
<p><b>/pwinst <i>InstallPwordOld</i></b></p> <p>Use pwinst to supply a previously defined installation-level password to the Integrity Client installation program. The following illustrates two variations of the pwinst installation command line switch:</p> <pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> [/additional switches...]</pre> <pre>iclientSetup_1101.exe /pwinst <i>InstallPwordOld</i> /pwinstset <i>InstallPwordNew</i></pre> <p>Default Value: Not applicable during initial installation.</p>

## The Configuration File Installation Command Line Specifier

Use the installation configuration file command line specifier to specify an optional installation configuration file to load when installation is completed. The following table lists the installation configuration file command line switch.



If used, the installation configuration file specifier must *not* be prefaced by a slash ("/") and *must* be the last switch on an installation command line.

The following table describe the installation configuration file command line specifier.

Configuration File Installation Command Line Switch
<p><b>"Path to Configuration File"</b></p> <p>Use the installation configuration file specifier to specify an installation configuration file to be loaded after installation has completed. The following illustrates the placement of the configuration file command line switch.</p> <pre>iclientSetup_1101.exe [/switches...] "C:\Full\path\to\Configuration.ini"</pre> <p>Do not confuse the installation configuration file specifier with the /policy switch. If used, the installation configuration file specifier:</p> <ul style="list-style-type: none"> <li>• Must not be used on the same installation command line as the /policy switch</li> <li>• Must <i>not</i> be prefaced by a slash mark ("/")</li> <li>• <i>Must</i> be the last switch on the command line</li> </ul> <p>The installation configuration file specifier:</p> <ul style="list-style-type: none"> <li>• Must be enclosed in quotation marks ("")</li> <li>• Can be any valid Windows filename, but must use the .ini filename extension</li> <li>• Can use Microsoft Windows Universal Naming Convention (UNC) of \\servername\sharename to refer to an installation configuration file located on a shared network resource</li> </ul> <p>When an installation configuration file is specified on a command line, Integrity Client ignores the Policy_Info section of the specified configuration file.</p>

## The Policy File Installation Command Line Switch

The /policy installation command line switch is used in conjunction with Integrity Server only. Do not use the policy installation command line switch with Integrity Desktop.

## Post-installation Configuration Files

Policy files are text-based files that specify a complete set of Integrity Desktop operational and security settings.

There are three basic strategies for using a policy file during installation. Ranging from simplest to most complex these strategies are permissive, connective, and standard:

- Permissive specifies an installation policy file that contains low (“permissive”) security settings so that a newly installed instance of Integrity Desktop can immediately use any and all network resources and programs it discovers after initial startup.
- Connective specifies a policy file that lists the specific network resources and programs that a newly installed instance of Integrity Desktop needs to use to retrieve a corporate security policy from a Web server.
- Standard specifies a comprehensive policy file that lists all the network resources and programs an instance of Integrity Desktop will use during normal operation.

Refer to the *Integrity Client Reference Guide* for details about policy file sections, parameters, and variables.

## Optional Installation Resources

You can enhance default installation processes with installation wrappers and deployment tools.

### Optional Installation Wrappers

Network administrators can use an installation packaging program—such as InstallShield, Wise, or WinZip Self-extractor— to deploy the Integrity Desktop installation program, installation command line, and installation policy file in a single package or “wrapper.”

### Optional Installation Deployment Tools

Network administrators can use standard program deployment tools, such as Microsoft’s System Management Server (SMS) or IBM Tivoli’s Business Systems Manager to configure and deploy Integrity Desktop to client computers not served by an Integrity Server.

## Changing an Existing Installation

After Integrity Desktop has been installed, use an operational command line, described in Chapter 5, “Using Operational Command Lines,” to change user-level or installation passwords, or to reload settings from a policy file.

## Using Operational Command Lines

After Integrity Desktop has been installed, use operational command line switches to:

- Reset existing user-level or installation-level passwords
- Apply an updated license key to an existing instance Integrity Desktop
- Cause Integrity Desktop to read new settings from a policy (configuration) file

### Types of Command Lines

Use operational command lines to change post-installation password and configuration settings of Integrity Desktop.

### Types of Command Lines

There are two distinct types of Integrity Client command lines

- Operational command lines, described in this chapter
- Installation command lines described in “Installation Command Lines,” on page 14”

The following table illustrates the primary differences between the two types of command lines.

Operational Characteristic	Installation Command Line	Operational Command Line
When used	During installation	After installation
Used with file	Integrity Client Installation program <i>iclientSetup_IXen.exe</i> . <sup>a</sup>	Integrity Client program file <i>iclient.exe</i> .
Parameter delimiter	Slash mark (“/”)	Dash (“-”)
Configuration file specifier	<ul style="list-style-type: none"><li>• Does not include a special preceding command line switch</li><li>• Path and file name specifier must be enclosed in quotation marks (“”)</li><li>• Must be the last switch on an installation command line</li></ul>	<ul style="list-style-type: none"><li>• Must be preceded by the <code>-config</code> command line switch</li><li>• Path and file name specifier must be enclosed in quotation marks (“”)</li><li>• Must be the last switch on an operational command line</li></ul>

a. Where *IX* equals *ID* for Integrity Desktop, *IF* for Integrity Flex, of *IA* for Integrity Agent, and *en* is the language code.

This rest of this chapter consists of two sections:

- ““Operational Command Lines,” in the following section
- “Creating and Running Operational Command Lines,” on page 41

# Operational Command Lines

Use operational command lines to:

- Set or change user-level or installation-level passwords
- Force Integrity Client to load an optional configuration or policy file

## Overview of Operational Command Lines

The following illustrates the general form of an Integrity Client operational command line (line break added for readability):

```
iclient.exe  
[-switch_1 -switch_2 ... -switch_n] [-config "C:\full\path\to\configuration.ini"]
```

The operational command line consists of three primary elements:

- `iclient.exe` is the name of the Integrity Client program.
- Optional command line switches, preceded by a dash ("-"), set new installation-level or user-level passwords, modify existing passwords, or specify a license key value.
- `-config C:\full\path\to\configuration.ini` specifies the path to an optional configuration file to be loaded by a previously installed instance of Integrity Client.

## The Configuration File Operational Command Line Switch

Special syntactic rules apply to the installation configuration file command line switch (`-config "C:\full\path\to\configuration.ini"` in the example in the preceding section). If specified in an installation operational command line, the `-config` switch:

- Must be the last switch on the command line, followed by the path name and file name of the configuration file
- Must be prefaced by a dash ("-")
- Must enclose the path name and filename in quotation marks ("")
- Can use Microsoft Windows' Universal Naming Convention (UNC) of `\\servername\sharename` to refer to a policy file located on a shared network resource

When the operational configuration file command line switch is used, Integrity Client ignores the `Policy_Info` section of the specified configuration file.

## Operational Command Line Switches

All operational command line switches are preceded by a dash ("-").


Integrity Client recognizes seven operational command line switches (six for Integrity Desktop). The following table groups the operational command line switches into four



General Operational Command Line Switches	
<b>-upgradekey</b>	
	<p>Use the <code>upgradekey</code> switch to specify an existing upgrade key. The following illustrates the general form of the <code>upgradekey</code> switch:</p> <pre>iclientSetup_1101.exe -upgradekey <i>upgradeKeyOld</i></pre> <ul style="list-style-type: none"> <li>• Use the <code>/upgradekey</code> installation command line switch, described on page 8, to specify an existing upgrade key during reconfiguration of an existing instance of Integrity Client.</li> <li>• Use the <code>/upgradekeyset</code> installation command line switch, described on page 8, to create a new upgrade key during initial installation.</li> </ul> <p>The upgrade key suppresses the dialogs that normally appear during reconfiguration or upgrade. Contrast this with the installation-level password which prevents anyone from uninstalling or upgrading Integrity Client without supplying the password.</p> <p>Default: No default value.</p>

## Set or Modify Passwords Operational Command Line Switches

Use the general operational command line switches group to set new user-level or installation-level passwords, or to supply existing passwords. The following tables list the four set or modify passwords operational command line switches.

Set or Modify Password Operational Command Line Switches	
<b>-passwdset</b> <i>UserPwordNew</i>	
	<p>Use <code>passwdset</code> to set a new user-level password.</p> <p>A user-level password:</p> <ul style="list-style-type: none"> <li>• Must be a minimum of 6 characters and a maximum of 31 characters, and can not contain spaces</li> <li>• Can only be set when no Integrity Client database files (“<code>.rdp</code>” file name extension) are present in the computer’s <code>C:\%windir%\Internet Logs</code> folder</li> </ul> <p>The following illustrates the general form of the <code>passwdset</code> operational command line switch:</p> <pre>iclient_1101.exe /passwdset <i>UserPwordNew</i></pre>
	<p>Zone Labs, Inc. recommends that a user-level password <i>not</i> be set. A user-level password prevents the end-user from responding to Integrity Client alerts and interferes with the application of centrally administered updates and changes.</p>
	<p>Default Value: No default value.</p>

Set or Modify Password Operational Command Line Switches
<b>-password</b> <i>UserPwordOld</i>
<p>Use the <code>password</code> switch to supply a previously defined user-level password to Integrity Client. The following illustrates the general form of the <code>password</code> operational command line switch:</p> <pre>iclient.exe -password <i>UserPwordOld</i></pre> <p>After installation, the <code>password</code> switch can be used in conjunction with <code>passwset</code> (described in the preceding table entry) to update an existing user-level password. In the following, <code>password</code> enables an existing user-level password to be modified:</p> <pre>iclient.exe -password <i>UserPwordOld</i> -passwset <i>UserPwordNew</i></pre> <p>Default: None.</p>

Set or Modify Password Operational Command Line Switches						
<b>-pwinstset</b> <i>InstallPwordNew</i>						
<p>Use <code>pwinstset</code> to set a new installation-level password. An installation-level password prevents unauthorized changes to an existing Integrity Desktop installation.</p> <ul style="list-style-type: none"> <li>• An installation-level password must be a minimum of 6 characters and a maximum of 31 characters, and can not contain spaces.</li> <li>• Installation-level passwords do not affect the user's ability to change his or her personal security settings.</li> </ul> <p>The following table inset illustrates three uses of the <code>pwinstset</code> operational command line switch.</p> <table border="1"> <thead> <tr> <th>No current installation-level password</th> </tr> </thead> <tbody> <tr> <td><code>iclient.exe -pwinstset <i>InstallPwordNew</i></code></td> </tr> <tr> <td> <ul style="list-style-type: none"> <li>• In this example <code>pwinstset</code> sets the installation-level password for the first time.</li> </ul> </td> </tr> <tr> <th>Changing an existing installation-level password</th> </tr> <tr> <td><code>iclient.exe -pwinst <i>InstallPwordOld</i> -pwinstset <i>InstallPwordNew</i></code></td> </tr> <tr> <td> <p>In this example:</p> <ul style="list-style-type: none"> <li>• <code>Pwinst</code> specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• <code>Pwinstset</code> changes the installation-level password</li> </ul> </td> </tr> </tbody> </table> <p>Installation-level passwords can be:</p> <ul style="list-style-type: none"> <li>• <i>Set</i> from the command line only during initial installation</li> <li>• <i>Changed</i> during reinstallation if the <code>pwinst</code> switch appears on the same installation command line to enable the change</li> </ul> <p>The <code>reset</code> switch, does <i>not</i> clear the installation password.</p> <p>Integrity Client provides no other methods for changing or updating an installation-level password.</p> <p>Default Value: No default value.</p>	No current installation-level password	<code>iclient.exe -pwinstset <i>InstallPwordNew</i></code>	<ul style="list-style-type: none"> <li>• In this example <code>pwinstset</code> sets the installation-level password for the first time.</li> </ul>	Changing an existing installation-level password	<code>iclient.exe -pwinst <i>InstallPwordOld</i> -pwinstset <i>InstallPwordNew</i></code>	<p>In this example:</p> <ul style="list-style-type: none"> <li>• <code>Pwinst</code> specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• <code>Pwinstset</code> changes the installation-level password</li> </ul>
No current installation-level password						
<code>iclient.exe -pwinstset <i>InstallPwordNew</i></code>						
<ul style="list-style-type: none"> <li>• In this example <code>pwinstset</code> sets the installation-level password for the first time.</li> </ul>						
Changing an existing installation-level password						
<code>iclient.exe -pwinst <i>InstallPwordOld</i> -pwinstset <i>InstallPwordNew</i></code>						
<p>In this example:</p> <ul style="list-style-type: none"> <li>• <code>Pwinst</code> specifies the existing installation-level password to enable a change to the installation-level password</li> <li>• <code>Pwinstset</code> changes the installation-level password</li> </ul>						

Set or Modify Password Operational Command Line Switches
<b>-pwinst</b> <i>InstallPwordOld</i>
<p>Use <code>pwinst</code> to supply a previously defined installation-level password to a previously installed instance of Integrity Client. The following illustrates two variations of the <code>pwinst</code> operational command line switch:</p> <pre>iclient.exe -pwinst <i>InstallPwordOld</i> [/additional switches...]</pre> <pre>iclient.exe -pwinst <i>InstallPwordOld</i> -pwinstset <i>InstallPwordNew</i></pre> <p>Default Value: None.</p>

## The -config Operational Command Line Switch

Use the `config` operational command line switch to direct a previously installed instance of Integrity Client to load a configuration file. The following table lists the `config` operational command line switch.



If used, the `config` operational command line switch must be prefaced by a dash ("-") and must be the last switch on an operational command line.

The following table describes the `config` operational command line switch.

Configuration File Operational Command Line Switch
<b>-config "Path to Configuration File"</b>
<p>Direct a previously installed instance of Integrity Client to load a configuration file. The following illustrates the placement of the configuration file command line switch.</p> <pre>iclient_1101.exe [/switches...] -config "C:\Full\path\to\Configuration.ini"</pre> <p>Do not confuse the <code>-config</code> operational command line switch with the <code>-policy</code> operational command line switch.</p> <p>If used, the <code>config</code> operational command line switch:</p> <ul style="list-style-type: none"> <li>• Must not be used on the same command line with the <code>policy</code> operational command line switch.</li> <li>• Must be prefaced with a dash ("-")</li> <li>• <i>Must</i> be the last switch on the command line</li> </ul> <p>The path and file name specifier used with the <code>config</code> switch:</p> <ul style="list-style-type: none"> <li>• Must be enclosed in quotation marks ("")</li> <li>• Can be any valid Windows filename, but must use the <code>.ini</code> filename extension</li> <li>• Can use Microsoft Windows Universal Naming Convention (UNC) of <code>\\servername\sharename</code> to refer to an installation configuration file located on a shared network resource</li> </ul> <p>After using <code>-config</code>, the Control Center does not display certain new settings until after Integrity Desktop has been restarted.</p> <p>When <code>config</code> is specified on a command line, Integrity Client ignores the <code>Policy_Info</code> section of the specified configuration file.</p>

## The Policy Operational Command Line Switch

Do not use the `-policy` operational command line switch in conjunction with Integrity Desktop. The `policy` operational command line switch can only be used with Integrity Agent or Integrity Flex.

## Creating and Running Operational Command Lines

There are three ways to run an Integrity Desktop operational command line:

- Type the command into the Windows Start menu's **Run** dialog box.
- Type the command line into the Start in text entry area of a Windows shortcut (".lnk" file name extension).
- Type the command line into a Windows batch file (".bat" file name extension).

## Creating an Operational Command Line Batch File

The following procedure illustrates how to type an operational command line into a Windows batch file.

### To type a command line into a Windows batch file:

- 1 In the **C:\Program Files\Zone Labs\Integrity Client** folder, right-click in an open area then choose **New | Text Document** from the shortcut menu.

Windows creates an empty text document named `New Text Document.txt`.

- 2 Rename the new text document to `FileName.bat`. The Integrity Desktop program accepts any valid Windows file name, but the `.bat` file name extension must be used.

- 3 Right-click on the newly created batch file then choose **Edit** from the pop-up menu.

Windows opens the batch file into the computer's default text editing program.

- 4 In the text editing window, type the following:

```
.\iclient.exe [-PwordInstallOld] [-PwordUserOld] -config ".\FileName.ini"
```

- Include the `-PwordInstallOld` switch only if an installation-level password was previously specified for this instance of the Integrity Desktop program.
- Include the `-PwordUserOld` switch only if a user-level password was previously specified for this instance of the Integrity Desktop program.
- `FileName.ini` is the name of an Integrity Desktop policy file. Enclose the file name in quotation marks (").

- 5 Save the changes and exit the text editing program.

Use the following procedure to run the completed batch file.

## Running an Operational Command Line Batch File

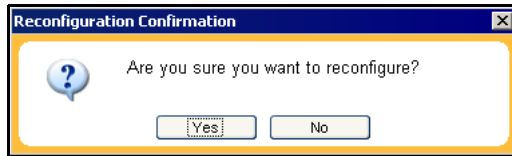
Complete the following procedure to run a batch file containing an Integrity Desktop operational command line.

### To run a command line batch file:

- 1 If it is running, shut down the Integrity Desktop program. In the Windows System Tray, right-click on the Integrity icon then choose **Shutdown Zone Labs Integrity...**

- 2 Double-click the batch file containing an operational command line.

Windows opens a command window and runs the batch file. If a silent installation has not been specified (“/s” installation command line switch), the Integrity Desktop program displays a Reconfiguration Confirmation dialog box.



**The Reconfiguration Confirmation dialog box**

- 3 In the **Reconfiguration Confirmation** dialog box, click **Yes**.

Windows closes the command window. The Integrity Desktop program reads the policy file specified by the command line contained in the batch file.

- 4 Restart the Integrity Desktop program.

The Integrity Desktop program loads the settings contained in the configuration file.

## Saving Configuration Files

Integrity Desktop provides a way to save current security and configuration settings into a text-based file. The file can then be used to apply the settings to other instances of Integrity Desktop either during or after installation.

### Saving Integrity Desktop Settings

Complete the following procedure to save Integrity Desktop's settings into a text-based file named policy.txt.

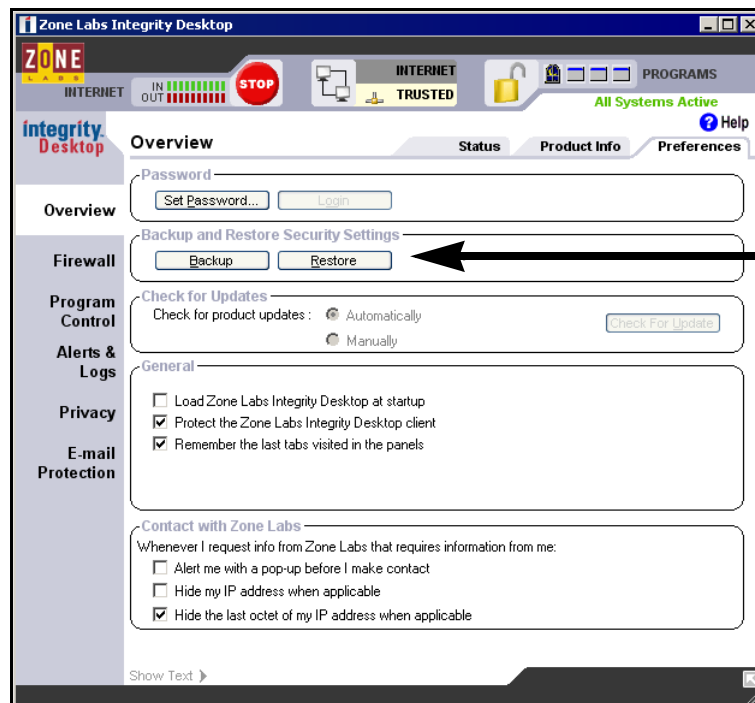


Integrity Desktop does not export all elements or attributes to an XML Policy file.

#### To view and save Integrity Desktop Settings:

- 1 In the Windows System Tray, double-click on the Integrity icon (shown at left) to open the Integrity Desktop program's Control Center.
- 2 In the Integrity Desktop Control Center, click to open the **Overview** panel then click the **Preferences** tab.

The Overview panel's Preferences tab appears.



For Integrity Desktop, use the **Backup and Restore** feature to save settings in an XML Policy file. (For Integrity Flex or Integrity Agent, use the Policies tab).

The Overview panel's Preferences Tab

- 3 In the **Preferences** tab, click **Backup**.

Integrity Desktop saves current configuration and security settings in a Zone Labs XML Policy file.

## Uninstalling Integrity Desktop

This chapter describes how to uninstall Integrity Desktop 4.0.


### Before You Begin

To complete the procedure in this chapter, you will need a copy of the Integrity Desktop uninstallation program `zauninst.exe`.

`Zauninst.exe` is normally placed into the default installation folder `C:\Program Files\Zone Labs\Integrity Client` by the installation program. If, however, the `install_log` installation command line switch, described on page 19, was used to specify a non-default pathname and folder for the installation log file, a separate copy of `zauninst.exe` must be obtained and copied to the installation folder in order to uninstall Integrity Desktop.

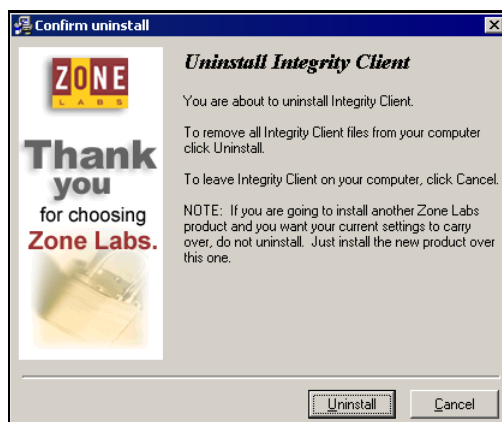
### Performing Prompted Uninstallation

Complete the following single-step procedure to have the Integrity Desktop uninstallation program prompt you during uninstallation.

-  **To perform a default uninstallation of Integrity Desktop:**
- Double-click the Integrity Desktop uninstallation program `zauninst.exe`.
- Zauninst.exe Icon** The uninstallation program starts. The uninstallation program prompts the user to confirm the uninstallation.

### Default Uninstallation

Run in its default state, the Integrity Desktop installation program performs a “clean” uninstallation: database files and registry entries are automatically removed from the computer.



The Select uninstall type dialog box

## Performing a Command Line Uninstallation

Version 3.5 of Integrity Client automatically performs a clean uninstallation. Because of this, Integrity Client version 3.5 no longer supports the `/clean` command line switch.

## Upgrading to Integrity Desktop

Zone Lab designed its Integrity family of programs for easy upgrade. However, some older Zone Labs products, particularly products that were not designed for use in an enterprise setting, have special upgrade requirements. For more information about specific upgrade issues see Chapter 3, “Upgrading to Integrity Desktop 4.0.”

Microsoft's Virtual Private Network (VPN) allows portable computers, such as laptop computers, to connect to a corporate network via a dial-up modem connection.

Unless properly configured, Integrity Desktop, like all Integrity products, will attempt to block the use of unrecognized programs or network entities necessary to create and use an VPN connection.

This chapter describes how to configure Integrity Desktop to work with most Windows VPN clients.

## Before You Begin

Integrity Desktop is compatible with the following versions of Microsoft Windows and VPN protocols.

## Supported Versions of Microsoft Windows

The following table lists the versions of Microsoft Windows operating systems that are compatible with Integrity Desktop:

Windows Operating System	Revision Levels
Windows 98	All
Windows ME	All
Windows NT 4.0	Service Packs (SP) 3, 4, 5, or 6
Windows 2000 Professional	SPs 1, or 2
Windows XP	All

## Supported VPN Protocols

The following table lists the VPN-related networking protocols that are recognized by Integrity Desktop.

Networking Protocol	Explanation and Comments
AH	Authentication Header protocol.
DES and 3DES	56-bit and 168-bit Data Encryption Standards.
ESP	Encapsulating Security Payload protocol.
GRE	Generic Routing Encapsulation protocol.

Networking Protocol	Explanation and Comments <i>(continued)</i>
H.323 RAS	H.323 Registration-Admission-Status. H.323. is an umbrella recommendation from the International Telecommunications Union (ITU) for multimedia communications over Local Area Networks (LANs).
IKE	Internet Key Exchange protocol.
IPSec	IP Security Protocol. Only the Windows 2000 VPN client automatically includes IPSec support.
L2F, L2TP	Layer 2 Forwarding protocol, Layer 2 Tunneling Protocol. L2TP is a more secure variation of PPTP.
LDAP	Lightweight Directory Access Protocol.
PPTP	Point-to-Point Tunneling Protocol.
RADIUS	Remote Authentication Dial-In User Service.
SKIP	Simple Key Management for Internet Protocol.
XAUTH	Extended Authentication.

## Overview of VPN Setup

Configuring Integrity Desktop to operate with a VPN consists of three major tasks:

- “Identifying Trusted Network Resources,” in the following section
- “Granting Programs Access,” on page 53
- “Enabling VPN Protocols,” on page 56

Each of these sections contain the steps necessary to complete each of these tasks.

## Identifying Trusted Network Resources

The first major task required to enable Integrity Desktop to operate with a VPN client is to add to the Trusted Zone all the network resources used by the VPN client.

Adding network resources to the Trusted Zone allows Integrity Desktop to automatically use these resources without blocking traffic or generating alerts.

## Types of Network Resources

For a given VPN configuration, there are two categories of network resources: required and optional. The following table lists the two categories of resource and describes how to define the resources to Integrity Client.

Type of Network Resource	Type of Integrity Desktop Network Element
Required VPN resources.	
VPN network equipment, such as: <ul style="list-style-type: none"> <li>• VPN Concentrators or Gateways</li> <li>• Corporate LANs that will be accessed by the VPN client computer</li> <li>• Corporate Wide-area network (WAN) subnets that will be accessed by the VPN client computer</li> </ul>	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address</li> <li>• Host / site</li> </ul>
Network resources accessed by the VPN client computer, such as: <ul style="list-style-type: none"> <li>• Computers</li> <li>• LANs, and WANs</li> </ul>	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address or IP Range</li> <li>• Host / Site</li> <li>• Subnet</li> </ul>
<ul style="list-style-type: none"> <li>• Programs or services required to provide VPN connectivity.</li> </ul>	In the <b>Program Control</b> panel's <b>Programs</b> tab: <ul style="list-style-type: none"> <li>• Programs used to provide VPN connectivity, including Windows services.</li> </ul>
VPN Resources that vary by installation	
Originating network resources, such as: <ul style="list-style-type: none"> <li>• A home network</li> <li>• Local client computer's NIC loopback address<sup>a</sup></li> </ul>	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address or IP Range</li> <li>• Subnet</li> </ul>

Type of Network Resource	Type of Integrity Desktop Network Element
DNS Servers	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address</li> <li>• Host / site</li> </ul>
Dynamic Host Control Protocol (DHCP) Servers	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address</li> <li>• Host / site</li> </ul>
Security servers such as: <ul style="list-style-type: none"> <li>• Remote Authentication Dial-In User Service (RADIUS)</li> <li>• Access Control Entry (ACE)</li> <li>• Terminal Access Controller Access Control System (TACACS+)</li> </ul>	Use the <b>Firewall</b> panel's <b>Zones</b> tab to identify the following types of network resources to Integrity Client: <ul style="list-style-type: none"> <li>• IP Address</li> <li>• Host / site</li> </ul>

a. Do not run proxy software on a client computer that has a local host loopback IP address of 127.0.0.1.

Some examples fall into more than one category depending on how they are used in a given network.

## Undesired Blocking of Remote Networks

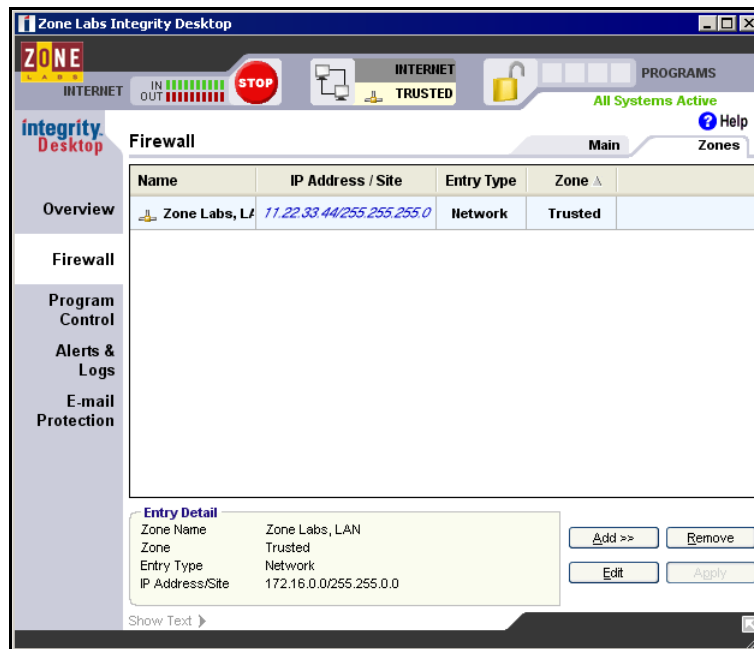
Integrity Desktop cannot identify networks on the remote side of a VPN connection. Traffic from these networks will be blocked if not explicitly added to the Trusted Zone.

Complete the following procedure to add network resources to the Trusted Zone.

### To add a network resource to the Trusted Zone:

- 1 Open the Firewall panel's Zones tab. In the Integrity Desktop Control Center, click **Firewall**, then click the **Zones** tab.

The Firewall panel's Zones tab appears.

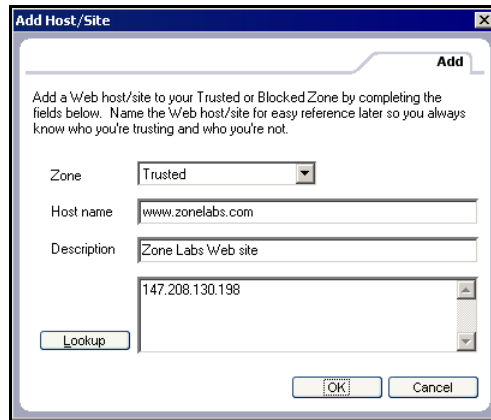


The Firewall panel's Zones tab

- 2 In the **Firewall** panel's **Zones** tab:
  - a In the list of networks right-click on an entry type.  
A shortcut menu appears.
  - b In the shortcut menu choose **Add**, then choose one of the four types of network resource listed under "Types of Network Resources," on page 50.
    - Host/Site
    - IP Address
    - IP Range

- Subnet

The dialog box for the type of network resource being added appears. The Add Host/Site dialog box is shown here.



Assign the network resources required by the VPN client to the Trusted Zone.

The Add Host/Site dialog box is one of four network resource dialog

- 3 In the **Add** dialog box:
  - a In the **Zone** list, choose **Trusted**.
  - b In the **Description** text entry area, type a free-form text description of the new resource.
  - c Type other IP address or host information as required.
- 4 Click **OK** to save your changes and return to the Firewall panel's Zones tab.  
Integrity Desktop adds the new network resource to the list of network resources.
- 5 Perform one of the following:
  - Repeat the procedure to add other network resource types to the Trusted Zone  
—or—
  - In the **Firewall** panel's Zones tab, click **Apply** to add the new network resources to the Integrity Desktop database.
- 6 Continue to the next section.

## Granting Programs Access

The second major task is to grant access to VPN client programs required for network access. For example, for VPNs that use Cisco VPN concentrators, Cisco provides the following three programs for use with its Series 3000 gateways:

- cvpnd.exe
- ipsecdialer.exe

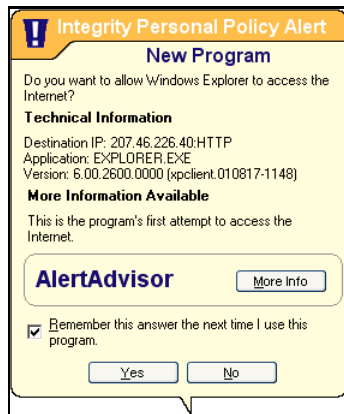
- xauth.exe

Integrity Desktop provides two ways to grant network to a program. Both of these are described in the following sections:

- ““Granting Program Access in Response to an Alert Box,” in the following section
- “Granting Program Access with the Control Center,” on page 54

## Granting Program Access in Response to an Alert Box

When a program requests network access for the first time, or if a program was not previously been granted access, Integrity Desktop displays a New Program alert box.



Click **Yes** to grant the program access. Check **Remember this program...** to have Integrity Desktop automatically grant this program network access the next time it requests it

A New Program alert box

When a New Program Alert box appears, perform the following steps to allow a program to access the network.

### To respond to a New Program Alert box:

- 1 If you want Integrity Desktop to automatically grant this program network access the next time it requests it, select the **Remember this answer the next time I use this program** check box.
- 2 Click **Yes** to grant the program access.

Integrity Desktop adds the program to the list in the Program Control panel's Programs tab.

## Granting Program Access with the Control Center

It is also possible to change the access permission of a program that is already recognized by Integrity Desktop.

## Hierarchy of Program Access Permissions

Integrity Desktop uses the following rules to apply access and server permissions to programs:

- A program cannot access the Trusted Zone without a corresponding access level to the Internet Zone.
- A program cannot have server permissions without corresponding access rights.

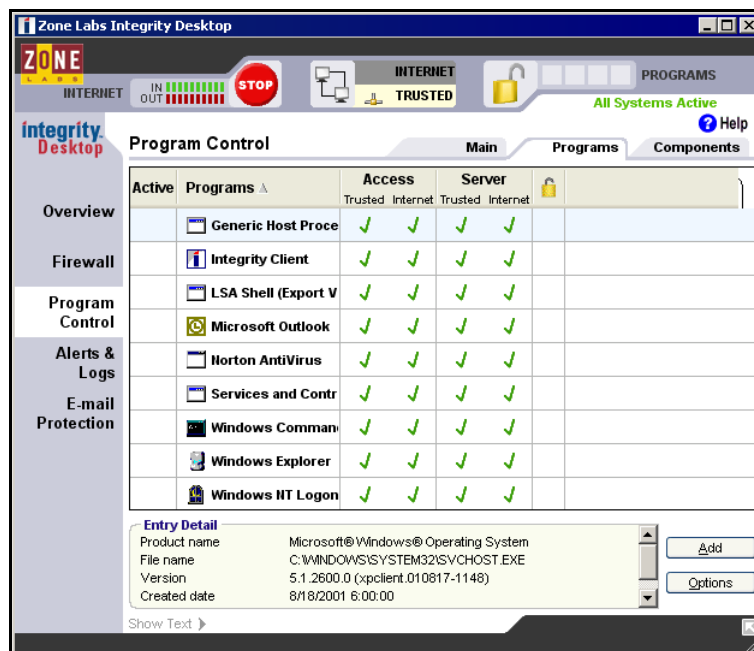
As permission choices are made, Integrity Desktop 4.0 automatically applies these rules to the other permissions categories.

Complete the following procedure to use the Control Center to grant a program access.

### To add trusted programs:

- 1 In the Integrity Desktop Control Center, click **Program Control**, then click the **Programs** tab.

The Program Control panel's Programs tab appears.



The Program panel's Programs tab

- 2 On the **Programs** tab, click **Add**.
- 3 To assign access permission to the new program, in the **Programs** tab:
  - a In the **Access** column, click one of the icons under **Trusted**.  
A shortcut menu appears



Use the shortcut menu to allow a program to act as a server.

- b In the menu choose **Allow**.  
Integrity Desktop places a green check mark under Trusted in the Server column.
- 4 Perform one of the following:

- Repeat the procedure to grant access to additional trusted programs  
—or—
- Continue to the next procedure, “”Enabling VPN Protocols”

## Enabling VPN Protocols

In its default configuration, Integrity Desktop allows specific VPN protocols through the firewall at high security. Use the following procedure to verify this setting.



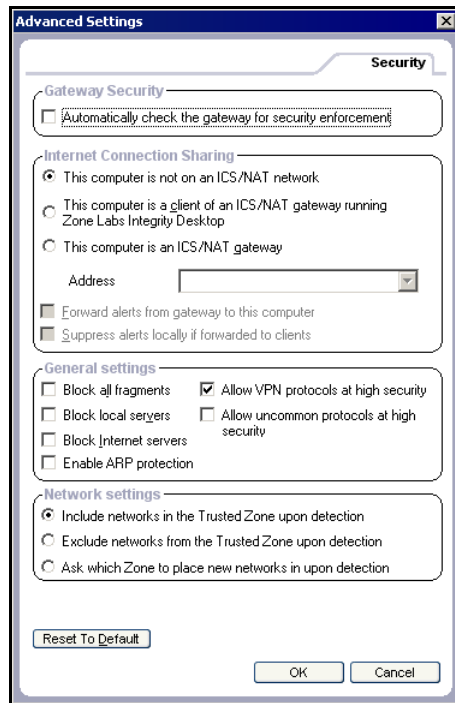
In a few cases specifying *BlockFragments* equal to *Yes* may interfere with the operation some of slow network connections, such as dial-up VPN connections.

### To enable specific VPN protocols through the Integrity Desktop firewall at high security:

- 1 Open the Advanced Settings dialog box:
  - a In the **Firewall** panel, click the **Main** tab.

- b In the Firewall panel's **Main** tab, in the **Internet Zone Security** area click the **Advanced** button.

The Advanced Settings dialog box appears.



Choosing **ALLOW VPN Protocols at high security** allows the AH, GRE, ESP, and SKIP protocols to transit the network at high security.

The Firewall panel's **Advanced Settings** dialog box

- 2 In the **Advanced Settings** dialog box, in the **General Settings** area, click to select the **Allow VPN protocols at high security** check box.
- 3 Click **OK** to save the changes and return to the Program panel's Programs tab.

This completes Integrity Desktop VPN setup.

## Using Troubleshooting Aids

The following sections describe three Integrity Desktop settings that can help when troubleshooting a newly installed VPN client:

- “Enabling Alert Logging,” in the following section
- “Automatically Assigning New Network Resources to the Trusted Zone,” on page 60
- “Enabling Program Learning Mode,” on page 60

The steps necessary to complete each of these tasks are contained in the following sections.

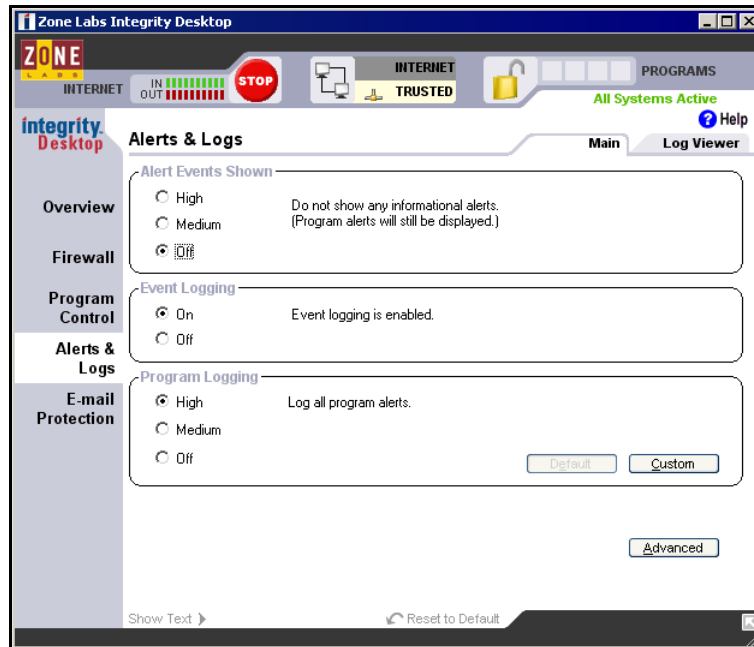
## Enabling Alert Logging

If problems occur, the Integrity Desktop alert log can provide useful troubleshooting information. Perform the following procedure enable alert logging.

### To enable Integrity Desktop alert logging:

- 1 In the Integrity Desktop Control Center, click **Alerts & Logs**, then click the **Main** tab.

The Alerts & Logs panel's Main tab appears.



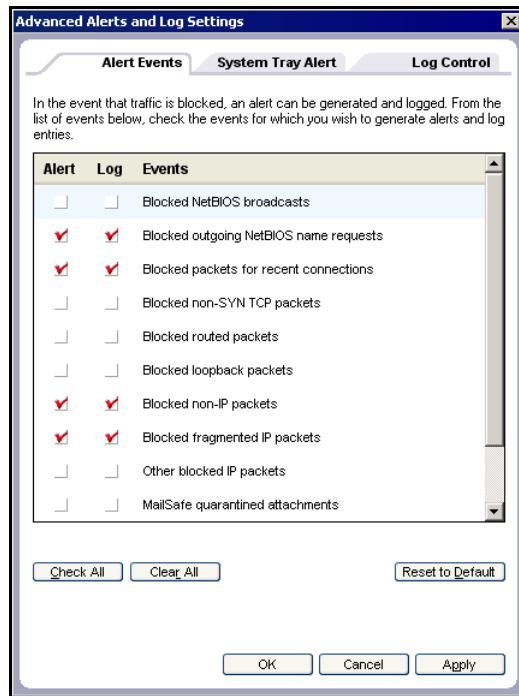
The Alerts & Logs panel's Main tab

- 2 In the **Main** tab, in the **Event Logging** area choose **On**.

Integrity Desktop enables alert logging.

- 3 In the **Main** tab, click **Advanced**.

The Advanced Alerts and Log Settings dialog box, Alert Event tab, appears.



The Advanced Alerts and Log Settings dialog box, Alert Events tab

- 4 In the Advanced Alerts and Log Settings dialog box:
- Click **Check All** to enable alerts for all blocked network traffic.
  - Click **OK** to return to the Alerts & Logs panel's Main tab.

## Viewing the Alert Log

Perform the following one-step procedure to view the Integrity Desktop alert log.

### To view the Integrity Desktop alert log:

- In the **Alerts & Logs** tab, click the **Log Viewer** tab.

The Integrity Desktop alert log appears.

## Automatically Assigning New Network Resources to the Trusted Zone

During troubleshooting, direct Integrity Desktop to automatically assign newly detected network resources to the Trusted Zone.



The following procedure applies low security settings to aid in troubleshooting. When troubleshooting has been completed, be sure to return the security settings to the levels necessary to protect the VPN client computer.

### To assign newly detected network resources to the Trusted Zone:

1 If it is not already open from the preceding procedure, open the Advanced Settings dialog box:

- a In the **Firewall** panel, click the **Main** tab
- b In the Firewall panel's **Main** tab click the **Advanced** button.

The Advanced settings dialog box appears. (Shown on page 57).

2 In the **Advanced Settings** dialog box, in the **Network Settings** area, click to select the **Include networks in the Trusted Zone upon detection** option button.

3 Click **OK** to save your changes and close the Advanced Settings dialog box.

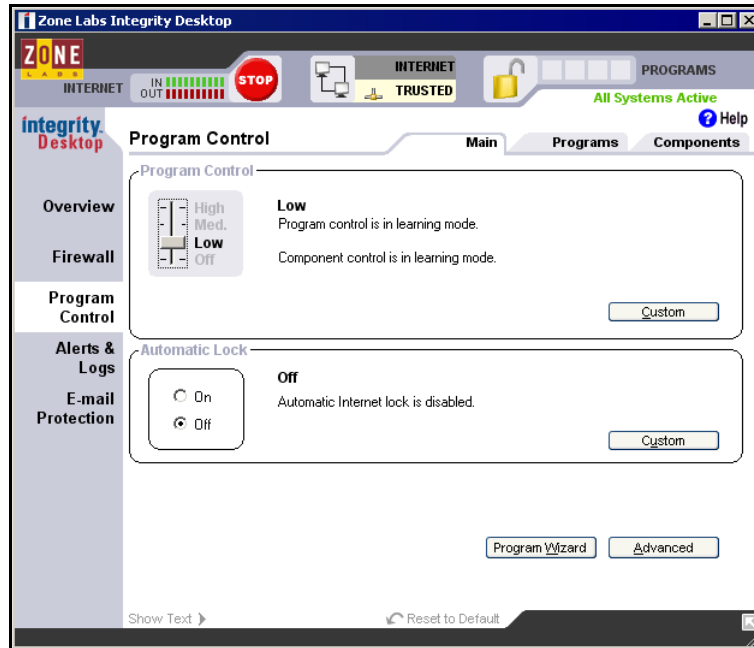
Integrity Desktop applies the changes to the database.

## Enabling Program Learning Mode

Place Integrity Desktop in learning mode to allow new programs to be added automatically to the list of recognized programs.

### To enable learning mode:

- 1 In the Integrity Desktop Control Center, click **Program Control**, then click the **Main** tab.  
The Program Control panel's Main tab appears.



The Program panel's Main tab

- 2 In the **Main** tab, in the **Program Control** area drag the program control slider to **Low**.

Integrity Desktop applies learning mode.

Be sure to return the Program Control slider to a more secure setting after completing troubleshooting of the VPN connection.

---

## Automatically Downloading Configuration Files

---

This chapter describes how to use the `autoconfig` section of a configuration file to automatically download (“pull”) updated security policies from a Web server at specified intervals.

### What’s New in this Release?

Beginning with product version 4.0, Integrity Desktop exports (writes) and imports (reads) configuration and security settings in a new Zone Labs XML Policy format (`.xml` filename extension). Because of this, this chapter now contains a description of the new Zone Labs XML Policy’s `autoconfig` element and attributes.

To ensure compatibility with previous releases, Integrity Desktop 4.0 retains the ability to read, but not write, older style configuration files (`.ini` filename extension). Because of this, this chapter also retains its description of the older style configuration file’s `autoconfig` section and parameters.

### Downloading Updated Configuration Files

In networks that are not equipped with Integrity Server, Integrity Desktop supports two ways of updating a client computer’s configuration:

- Manually download a configuration of XML Policy file to a computer, then use command line parameters to force Integrity Desktop to read the file.
- Direct Integrity Desktop to periodically download (“pull”) a configuration of XML Policy file from a network server and read the file.

Both of these methods differ from the deployment (“push”) of enterprise security policies used in networks equipped with Integrity Server. See the Zone Labs *Administrator Guide* for more information about enterprise security policy configuration and deployment.

### Before You Begin

Automatic downloading of configuration files requires the following two components:

- A properly configured Web server
- An XML Policy file or older style configuration file

Each of these components are described in the following sections.

## Web Server Requirements

Unlike automatic alert log uploading, which requires Microsoft IIS, automatic downloading of configuration files can be performed from any standards-based Web server.

## Configuration or XML Policy File Requirements

In an old style configuration file, the `autoconfig` configuration file section specifies from where and how often to download configuration files that are stored on a web server.

In the new Zone Labs XML Policy file, the `autoconfig` element's attributes specify from where and how often to download configuration files that are stored on a web server.

Procedures for configuration both types of file are contained in the following section.

## Configuring the *autoconfig* Element or Section

Beginning with product version 4.0, Integrity Desktop exports (writes) and imports (reads) configuration and security settings in a new Zone Labs XML Policy format (*.xml* filename extension). To ensure compatibility with previous releases, Integrity Desktop 4.0 also retains the ability to read, but not write, older style configuration files (*.ini* filename extension).

This section contains two procedures:

- “Editing the `autoconfig` Element in an XML Policy File,” in the following section
- “Editing `autoconfig` Configuration File Parameters,” on page 68.

Both procedures provide a step-by-step description of the steps necessary to specify automatic configuration file download settings in an XML Policy file or configuration file.

## Editing the *autoconfig* Element in an XML Policy File

The Zone Labs XML Policy contains a complete set of security and environment elements and attributes. In networks that do not have Zone Labs Integrity Server, use the `autoconfig` element and its attributes contained in an XML Policy file to schedule periodic downloads of updated security policies.

In the following example, the `autoconfig` section of a configuration file specifies the download of a configuration file named *policy.xml*. Integrity Desktop downloads the file to, and reads the downloaded file from, the Integrity Desktop installation folder (the folder containing the program file *iclient.exe*).

```
/ZoneLabsSettings/configuration
<configuration ... />
  <autoconfig autoCheck="true" source="machine.company.com/pathName/policy.xml"
frequency="boot" timeout="60" retries="3" retryInterval="3" lastDownloadTime="invalid date"/>
  ...
```

The following section describes each of the `autoconfig` element's attributes in detail.

## List of *autoconfig* Configuration File Parameters

The following table lists the *autoconfig* element's attributes.

/ZoneLabsSettings/configuration/autoconfig		
Attribute	Type, Values and Description <i>(Sheet 1 of 2)</i>	
<b>autoCheck</b> Enable periodic downloading of configuration files.	<b>Type:</b>	boolean
	<b>Recognized values:</b>	true, <u>false</u>
	Specify the autoCheck attribute equal to true to enable the automatic, periodic downloading of configuration files from a Web server.  The autocheck element must be <i>true</i> to enable the other autoconfig attributes described in this table.	
<b>frequency</b> Specify how often to download an XML Policy or configuration file.	<b>Type:</b>	integer
	<b>Recognized values:</b>	boot, daily, weekly, integer number of minutes.
	Use the frequency attribute to specify how often to seek a new configuration file at the address specified by the source parameter, described later in this table. <ul style="list-style-type: none"> <li>• Any integer value is interpreted as the number of minutes.</li> <li>• Integrity Client can not check for an updated configuration file more frequently than the end-point computer is re-booted.</li> </ul> The autocheck element, described earlier in this table, must be true to enable the frequency attribute.	
<b>lastDownloadTime</b> Read-only time of last successful download.	<b>Type:</b>	Read-only formatted string.
	<b>Displayed value:</b>	String formatted as <i>yyyy-mm-dd_hh:mm:ss</i>
	Integrity Desktop automatically updates the value of the lastDownloadtime attribute at the time a successful download has completed.  The autocheck element, described earlier in this table, must be true to enable the updating of the lastDownloadtime attribute.	
<b>retries</b> Specify how many times to attempt a download.	<b>Type:</b>	integer
	<b>Recognized values:</b>	Integer number of retry attempts.
	Use the retries attribute to specify how many times Integrity Desktop will attempt to complete an unsuccessful download. <ul style="list-style-type: none"> <li>• The retries attribute operates in conjunction with the retryInterval attribute described in the following attribute table entry, and the timeout attribute, described later in this table.</li> <li>• The autocheck element, described earlier in this table, must be true to enable the retries attribute.</li> </ul>	

<b>/ZoneLabsSettings/configuration/autoconfig</b> (continued)	
<b>Attribute</b>	<b>Type, Values and Description</b> (Sheet 2 of 2)
<b>retryInterval</b> Specify how long to wait between unsuccessful download attempts.	<b>Type:</b> integer
	<b>Recognized values:</b> Integer number of seconds.
	Use the <code>retryInterval</code> attribute to specify how many seconds to wait between unsuccessful download attempts. <ul style="list-style-type: none"> <li>The <code>retryInterval</code> attribute operates in conjunction with the <code>retries</code> attribute described in the preceding table entry, and the <code>timeout</code> attribute described later in this table.</li> <li>The <code>autocheck</code> element, described earlier in this table, must be true to enable the <code>retryInterval</code> attribute.</li> </ul>
<b>source</b> Specify the source of configuration or XML Policy files.	<b>Type:</b> Formatted string
	<b>Recognized values:</b> Valid URL or IP address of a Web server containing configuration of XML Policy files.
	Use the <code>source</code> attribute to specify the location of the Web server containing configuration or XML Policy files.  The <code>autocheck</code> element, described earlier in this table, must be true to enable the <code>source</code> attribute.
<b>timeout</b> Specify how long to wait before abandoning an unsuccessful download attempt.	<b>Type:</b> integer
	<b>Recognized values:</b> Integer number of seconds.
	Use the <code>retries</code> attribute to specify how many seconds Integrity Desktop will wait before abandoning a download attempt. <ul style="list-style-type: none"> <li>The <code>timeout</code> attribute operates in conjunction with the <code>retries</code> attribute, and the <code>retryInterval</code> attribute, both described earlier in this table.</li> <li>The <code>autocheck</code> element, described earlier in this table, must be true to enable the <code>timeout</code> attribute.</li> </ul>
<b>Level 4 Child Elements</b>	
None	The <code>autoconfig</code> element contains no child elements.

See also the XML Policy Reference for a complete description of the Zone Labs XML Policy.

## Editing *autoconfig* XML Policy Attributes

Complete the following procedure to type `autoconfig` attributes into an XML Policy file.

### XML and Case-sensitivity

Unlike older style configuration files, XML is case sensitive: `autoconfig` (the correct XML Policy capitalization) is not the same as `AutoConfig`, `autoConfig`, or any other variation of upper and lower case characters. Be sure to capitalize XML elements and attributes as written in the following procedure.

### To type *autoconfig* parameters into an XML Policy file:

- 1 Open the XML Policy file into a text-editing program. The default text-editing program for Windows is Notepad although you may prefer to use an editor designed to work with XML elements and attributes.
- 2 In the XML Policy file, locate the running ruleset's *autoconfig* element:
  - a Search for or browse to the ruleset that contains a name="runningruleset" attribute as shown in the following example:
 

```
<ruleset name="runningruleset" start="afterstartup" stop="onshutdown">
```

Only the running ruleset's *autoconfig* element will properly configure automatic configuration file download.
  - b From the running ruleset element, search for or browse forward (downward) in the XML Policy file for the *autoconfig* child element. The following illustrates the general form of the *autoconfig* element (line break added):
 

```
<autoconfig autoCheck="true" source="machine.company.com" frequency="boot"
  timeout="60" retries="3" retryInterval="3"/>
```
- 3 In the XML Policy file, add or modify the *autoconfig* element's attributes, including the quotation marks:
  - a Type autoCheck="true"
  - b Type source="http://Web\_Server\_Address/Configurations/FileName.xml"
  - c Type frequency="[boot, daily | weekly | *Integer number of minutes*]"
  - d Optionally, type values for timeout, retries, and retryInterval.
- 4 Save the changes and exit the text-editing program.

After adding the *autoconfig* parameters to the client computer's XML Policy file, use an operational command line to direct Integrity Desktop on the client computer to read the file.

## Editing the *autoconfig* Section in a Configuration File

A security policy contains a complete set of security and environment variables. In networks that do not have Zone Labs Integrity Server, use the *autoconfig* section of a configuration file to schedule periodic downloads of updated security policies.

In the following example, the *autoconfig* section of a configuration file specifies the download of a configuration file named *policy.ini*. Integrity Desktop downloads the file to, and reads the downloaded file from, the Integrity Desktop installation folder (the folder containing the program file *iclient.exe*).

```
[autoconfig]
Autocheck={Yes | No}
Source=http://ServerAddress/Configurations/Configuration.ini
Frequency={Boot | Daily | Weekly | Integer Number of Minutes}
Timeout=IntegerNumberOfSeconds
Retries=IntegerValue
RetryInterval=IntegerNumberOfSeconds
```

The following section describes each of the autoconfig section's parameters in detail.

## List of *autoconfig* Configuration File Parameters

The following table lists the six parameters recognized by the autoconfig section of a configuration file.

<b>autoconfig Parameter</b>	<b>Description</b>
<b>Autocheck</b>	<p>Use Autocheck to enable or disable automatic downloading of configuration files from a Web server.</p> <ul style="list-style-type: none"> <li>• Type <code>Autocheck=Yes</code> to enable automatic downloading</li> <li>• Type <code>Autocheck=No</code> to disable automatic downloading</li> </ul> <p>Specify <code>AutoCheck</code> equal to <code>Yes</code> to enable the remaining autoconfig parameters.</p>
<b>Frequency</b>	<p>Use Frequency to specify the how often Integrity Desktop checks for updated config.ini files.</p> <p>The following example command line illustrates the general form of the Frequency parameter:</p> <pre>Frequency=[<i>Boot, Daily   Weekly</i>   IntegerNumberOfMinutes]</pre> <ul style="list-style-type: none"> <li>• <code>Boot</code> attempts to download a configuration file each time Integrity Desktop (as distinguished from the computer) is started</li> <li>• An integer value specifies a number of minutes. Interval must be greater than 60 minutes.</li> </ul>
<b>Retries</b>	<p>Use Retries to specify the number of times Integrity Desktop will attempt to load an configuration file after an original download attempt fails.</p> <p>In the following example, Integrity Desktop tries 3 times to download a file after an initial failure</p> <pre>Retries=3</pre> <p>If, after the third attempt, Integrity Desktop is not successful, the next download attempt will occur as determined by the value of Frequency, above.</p>

<b>autoconfig Parameter</b>	<b>Description</b> <i>(continued)</i>
<b>RetryInterval</b>	<p>Use <code>RetryInterval</code> to specify the number of seconds Integrity Desktop waits after a failed download before trying again.</p> <p>In the following example, Integrity Desktop waits 600 seconds (10 minutes) after an initial failure before making another attempt to download the file.</p> <pre>RetryInterval=600</pre>
<b>Source</b>	<p>Use <code>Source</code> to specify the URL of the remote configuration file. Both standard HTTP and HTTPS (Secure Socket Layer) transfers are supported.</p> <p>The following two examples illustrate the <code>Source</code> parameter:</p> <pre>source=http://111.111.111.111/Configuration/config.ini</pre> <pre>source=https://MyServer.com/Configuration/config.ini</pre> <p>In these examples, Integrity Desktop downloads a configuration file from IP address 111.111.111.111 or from the server named MyServer.</p> <ul style="list-style-type: none"> <li>• After Integrity Desktop uploads an updated configuration file, it enforces the security configuration defined by that file immediately.</li> <li>• When https is used, the web server may require user authentication such as a password, depending on the type of authentication method used on the server.</li> </ul>
<b>Timeout</b>	<p>Use <code>Timeout</code> to specify the number of seconds to wait before abandoning a scheduled download.</p> <p>In the following example, Integrity Desktop waits 60 seconds for a scheduled file download to commence or resume.</p> <pre>Timeout=60</pre> <p>If a scheduled download can not be completed within the timeout period, then Integrity Desktop:</p> <ul style="list-style-type: none"> <li>• Abandons the download</li> <li>• Waits for the value of the <code>RetryInterval</code> parameter to expire</li> <li>• Attempts to complete an interrupted download as many times as specified by the <code>Retries</code> parameter</li> <li>• Abandons the download until the next interval specified by the <code>Frequency</code> parameter.</li> </ul>

## Editing *autoconfig* Configuration File Parameters

Complete the following procedure to type *autoconfig* parameters into a configuration file.

### To type *autoconfig* parameters into a configuration file:

- 1 Open the configuration file into a text-editing program. The default text-editing program for Windows is Notepad.
- 2 Perform one of the following:
  - In the text editing window, locate the *autoconfig* section of the configuration file

—or—

- If the configuration file does not contain an `autoconfig` section, type `[autoconfig]`
- 3 In the configuration file, below the `autoconfig` section heading:
    - a Type `AutoCheck=Yes`
    - b Type `Source=http://Web_Server_Address/Configurations/FileName.ini`
    - c Specify a value for Frequency.
    - d Specify a value for Timeout, Retries, and Retry Interval.
  - 4 Save the changes and exit the text-editing program.

After adding the `autoconfig` parameters to the client computer's configuration file, use an operational command line to direct Integrity Desktop on the client computer to read the file.

## Updating with Zone Labs Integrity Server

As stated earlier, this chapter describes how to use the `autoconfig` section of a configuration file to periodically download (“pull”) a security configuration from a Web server to Integrity Desktop.

If your local network is equipped with Zone Labs Integrity Server, use Policy Studio to configure and deploy (“push”) enterprise policies to either Integrity Agent or Integrity Flex. Refer to the *Integrity Server Administrator Guide* for detailed information about configuring and deploying enterprise security policies to these other types of Integrity Client.

---

## Automatically Uploading Alert Logs

---

As it protects a computer, Integrity Desktop stores alerts in encrypted database files (.*rdb* file name extension).

The `AlertLog` section of a policy file can be used to cause Integrity Desktop to periodically archive alerts in text-based files. The `autouploadlog` section of a configuration or XML Policy file can then be used to periodically upload the archived alert log data to a Windows IIS server.

This chapter describes how to set up Integrity Desktop to periodically upload alert logs to a Windows IIS server.

### What's New in this Release?

Beginning with product version 4.0, Integrity Desktop exports (writes) and imports (reads) configuration and security settings in a new Zone Labs XML Policy format (.*xml* filename extension). Because of this, this chapter now contains a description of the new Zone Labs XML Policy's `autoconfig` element and attributes.

To ensure compatibility with previous releases, Integrity Desktop 4.0 retains the ability to read, but not write, older style configuration files (.*ini* filename extension). Because of this, this chapter also retains its description of the older style configuration file's `autouploadlog` section and parameters.

### Uploading Archived Alert Logs

In networks that are not equipped with Integrity Server, Integrity Desktop provides a mechanism for uploading archived alert logs. This method of uploading archived alert logs differs from the automatic collection of alert data from Integrity Flex or Integrity Agent performed in networks equipped with Integrity Server.

See the Zone Labs *Administrator Guide* for more information about Integrity Server and enterprise alert log collection.

### Before You Begin

Automatic uploading of Integrity Desktop alert logs requires two software components:

- Installation and configuration of Microsoft Internet Information Services (IIS) on the server that will receive the uploaded alert logs.
- A copy of Zone Labs, Inc.'s Active Server Page file *IDLogUpload.asp*.

Both of these required components are described below.

## Microsoft IIS

The first component required for automatic alert log uploading only is Microsoft Windows' Internet Information Services (IIS).

### IIS Configuration Guidelines

The IIS server assigned to receive uploaded alert logs:

- Does not require its own domain.
- Does not need to be dedicated entirely to log uploading. Ensure, however, that running IIS does not conflict with other services and shared server requirements in the network.
- Should not be placed in a network DMZ<sup>1</sup> for both security and traffic throughput reasons.
- Must be accessible to all client workstations that perform alert log uploading.

### IIS and Data Transfer Protocols

Integrity Desktop alert log upload uses either HTTP or HTTPS formats to upload archive log files. The following table lists the primary differences between the two modes as they apply to alert log uploading.

Transfer Protocol	Summary
HTTP	<p>HTTP (Hypertext Transfer Protocol):</p> <ul style="list-style-type: none"> <li>• Transfers data in plain-text format</li> <li>• Does not require client computers to accept Security Certificates</li> </ul> <p>When the destination IIS server is used for purposes other than receiving uploaded alert logs, and is also configured to use SSL, it is necessary to create a virtual directory to enable HTTP.</p>
HTTPS	<p>HTTPS (Hypertext Transfer Protocol Secure):</p> <ul style="list-style-type: none"> <li>• Uses Secure Socket Layer (SSL) to encrypt data</li> <li>• Requires a Certificate Authority</li> </ul> <p>For alert log upload it is recommended, but not required, to use a stand-alone root Certificate Authority.</p>

### IIS and Security Certificates

IIS can run on both the standard and server versions of Windows. Note, however, that certain standard versions of Windows—such as Windows 2000 Professional—may not properly support the stand-alone Certificate Authority feature used to upload Integrity Desktop alert logs in secure (HTTPS) mode.

If you are unfamiliar with Web and IIS server administration—including SSL certificates, DNS resolution and security patching—Zone Labs, Inc. recommends that you obtain professional assistance installing and configuring IIS.

1. When used in the context of data networking, the term DMZ describes equipment positioned between a network's internal and external defenses.

## IIS and Passwords

HTTPS authentication may also require password protection. Though it is prudent to employ passwords, they can interfere with automatic log upload.

## ZALogUpload Active Server Page

The second component required for automatic upload of log files is the Active Server Page *IDLogUpload.asp*. Obtain *IDLogUpload.asp* from:

- The Integrity Desktop Web site at <https://pt01.zonelabs.com/zapid>
- Your Zone Labs Systems Engineer or sales representative

After installing and configuring IIS and acquiring a copy of *IDLogUpload.asp*, proceed to the next section to enable and configure automatic alert log upload.

## Configuring the *autouploadlog* Element or Section

Installing and configuring Integrity Desktop to perform automatic alert log uploading consists of three primary tasks:

- ““Creating and Configuring a Destination Folder on the IIS Server Computer,” in the following section
- “Enabling Alert Log Archival,” on page 74
- “Configuring the autouploadlog Element or Section,” on page 77

Perform each of these tasks in the order listed to configure alert log upload.

### Task 1: Creating and Configuring a Destination Folder on the IIS Server Computer

The first task required by alert log uploading is to configure the destination folder on the IIS server computer that will receive uploaded alert logs.

## Managing Microsoft IIS Security

Operated in its default mode, Microsoft IIS may not provide sufficient levels of system security.

A complete discussion of IIS security is beyond the scope of this document. Within the context of uploading archived alert logs, the following general guidelines can help improve IIS security:

- Assign Windows folders special access permissions to enable only Integrity Desktop to create log files in the destination folder.

- Create a special username, such as *ZALogAdmin*, for alert log uploading and assign the name the necessary access permissions.

For more information about IIS, consult:

- Your computer's online help
- The Microsoft Web site at <http://www.microsoft.com/WindowsServer2003/iis/default.msp>

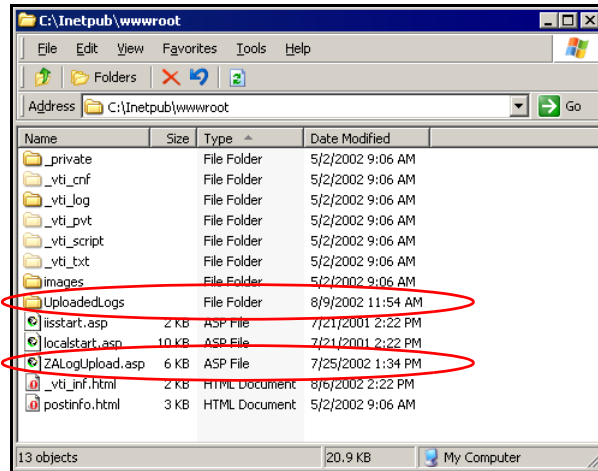
Whatever steps you take to improve the default security of Microsoft IIS, periodically review your system's security status to help prevent misuse or exploitation of IIS security behaviors.

### To create and configure the server's destination folder:

- 1 On the IIS server computer, create the destination folder `UploadedLogs`. On the server that will receive uploaded alert logs:
  - a Find the `C:\inetpub\wwwroot` folder.
  - b In the **wwwroot** folder, create a sub-folder named `UploadedLogs`.  
After these steps have been completed, the full path will on the server computer will be `C:\inetpub\wwwroot\UploadedLogs`
- 2 Enable **Write** permissions for the `UploadedLogs` folder. In the `C:\inetpub\wwwroot` folder:
  - a On the server computer, right-click the **UploadedLogs** folder then select **Properties** from the shortcut menu.  
The `UploadedLogs` Properties dialog box appears.
  - b In the `UploadedLogs` Properties dialog box, click the **Security** tab.
  - c In the **Security** tab, in the **Group or user names** area choose **Everyone**.
  - d In the **Security** tab, in the **Permissions for Everyone** area under **Allow**, click to select the **Write** check box.
- 3 Click **Apply**, then click **OK** to close the `UploadedLogs` Properties dialog box.  
Windows applies the write permission to the `UploadedLogs` folder.
- 4 Create a new Virtual Directory and assign Read and Write permissions to it.
  - a In the Windows **Start** menu, in **Administrative Tools** choose **Internet Service Manager**.
  - b In the Internet Service Manager dialog box, right-click on the entry for the local machine, then in the shortcut menu choose **New | Virtual directory**.
  - c In the **New Virtual Directory** dialog box, create a new virtual directory named `UploadedLogs` and map the directory to the `\Intepub\wwwroot\UploadedLogs` folder created in step 1, above.
  - d Click **OK** to apply the changes and return to the **Internet Service Manager** dialog box.
  - e In the **File** menu, choose **Close** to close the **Internet Service Manager** dialog box.

- 5 On the server computer, copy the `IDLogUpload.asp` file to the `c:\inetpub\wwwroot` folder.

After completing this step, both the `UploadedLogs` folder and the `IDLogUpload.asp` file are in the server computer's `C:\Inetpub\wwwroot` folder.



The `C:\inetpub\wwwroot` folder contains the `UploadedLogs` folder and the `IDLogUpload.asp` file

## Specifying Non-standard Folder Names

If you specify a path and subdirectory naming convention different than the one used in the preceding procedure, you must also change the value of `RemoteRelativeDirectory` in `IDLogUpload.asp` to the non-standard path name.

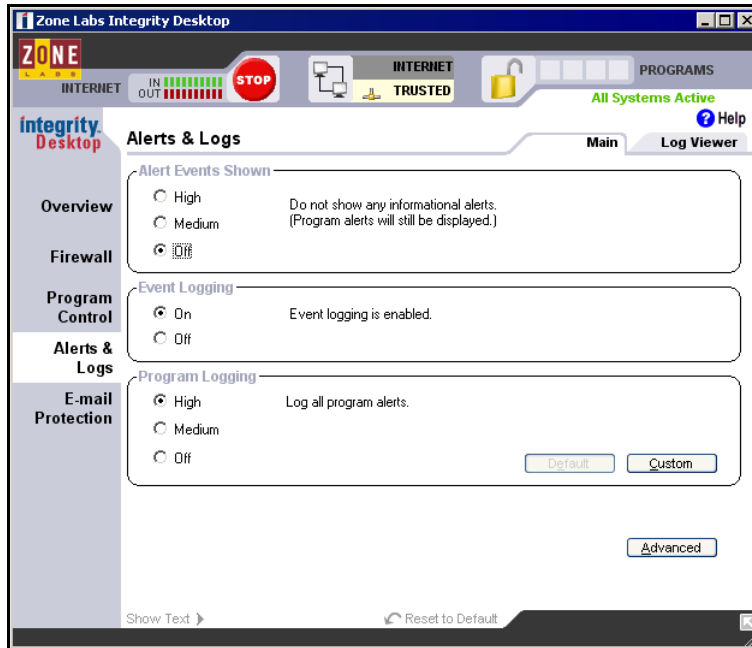
Proceed to ““Enabling Alert Log Archival,” in the following section.

## Task 2: Enabling Alert Log Archival

The second task required to perform alert log uploading is to enable alert log archival.

### To enable alert log archival:

- 1 In the Integrity Desktop Control Center, click **Alerts & Logs**, then click the **Main** tab.  
The Alerts & Logs panel's Main tab appears.

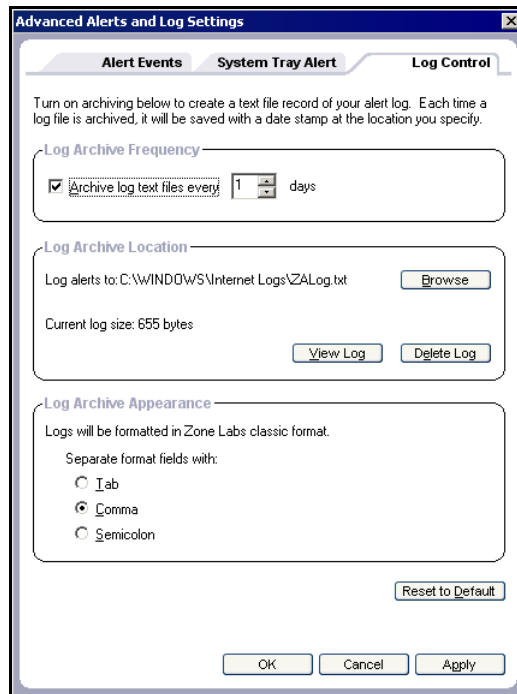


The Alerts & Logs panel's Main tab

- 2 In the **Alerts & Logs** panel's **Main** tab:
  - a in the **Event Logging** area, choose **On** to enable event logging.
  - b In the **Program Logging** area, choose **Medium** or **High** to enable program logging.

- 3 In the **Alerts & Logs** panel's **Main** tab, click the **Advanced** button then click the **Log Control** tab.

The Advanced Alerts and Logs Settings dialog box, Log Control tab appears.



The Advanced Alerts and Log Settings dialog box, Log Control tab

- 4 In the **Advanced Alerts and Logs** dialog box's **Log Control** tab:
  - a In the **Archive Log text files every** box, check the **Archive Log text files every** check box.
  - b In the **Log Archive Frequency** area, in the **Archive Log text files every** box choose the interval in days to archive alerts before uploading.
  - c Zone Labs, Inc. recommends leaving the alert log folder set to its default value. To change the default folder location for Integrity Desktop archived alert logs, In the **Log Archive Location** area, click **Browse** then choose a folder to store local copies of Integrity Desktop log files. The default folder location for Integrity Desktop log files is `C:\Windows\Internet Logs\`.
  - d To customize the delimiter parameter, in the Log Archive Appearance area choose **Tab**, **Comma**, or **Semicolon**.
- 5 Click **Apply** to save any changes, then click **OK** to return to the Alerts & Logs section's Main tab.

Integrity Desktop saves the changes in the database.

Proceed to ““Configuring the autouploadlog Element or Section,” in the following section.

## Task 3: Configuring the *autouploadlog* Element or Section

The third and final task required to perform alert log uploading is to manually edit the `autouploadlog` section of the client computer's XML Policy or configuration file.

Beginning with product version 4.0, Integrity Desktop exports (writes) and imports (reads) configuration and security settings in a new Zone Labs XML Policy format (*.xml* filename extension). To ensure compatibility with previous releases, Integrity Desktop 4.0 also retains the ability to read, but not write, older style configuration files (*.ini* filename extension).

This section contains two procedures:

- “Editing the `autouploadlog` Element in an XML Policy File,” in the following section
- “Editing the `autouploadlog` Section in a Configuration File,” on page 80.

Both procedures provide a step-by-step description of the steps necessary to specify automatic uploading of archived alert data in an XML Policy file or configuration file.

### Editing the *autouploadlog* Element in an XML Policy File

The Zone Labs XML Policy contains a complete set of security and environment elements and attributes. In networks that do not have Zone Labs Integrity Server, use the `autouploadlog` element and its attributes contained in an XML Policy file to schedule periodic uploads of archived alert log data.

In the following example, the `autouploadlog` element of an XML Policy file specifies path to the Active Server Page (ASP) file `IDLogUpload.asp`, as well as specifying the upload frequency, time, and retry intervals.

```
/ZoneLabsSettings/configuration
<configuration ... />
  <autouploadlog enabled="true" host="IIS Server_Address/IDLogUpload.asp" frequency="hour"
    action="archive" timeout="60" retries="3" retryInterval="3"/>
  ...
```

The following section describes each of the `autouploadlog` element's attributes in detail.

## List of *autouploadlog* Configuration File Parameters

The following table lists the *autouploadlog* element's attributes.

/ZoneLabsSettings/configuration/autouploadlog	
Attribute	Type, Values and Description <i>(Sheet 1 of 2)</i>
<b>enable</b> Enable periodic uploading or archived log files.	<b>Type:</b> boolean
	<b>Recognized values:</b> true, <u>false</u>
	Specify the enable attribute equal to true to enable the automatic, periodic uploading of archived log files to a Microsoft IIS server.  The enable element must be <i>true</i> to enable the other autouploadlog attributes described in this table.
<b>frequency</b> Specify how often to upload archived log files.	<b>Type:</b> integer
	<b>Recognized values:</b> boot, daily, weekly, integer number of minutes.
	Use the frequency attribute to specify how often Integrity Desktop uploads archived log files to the Windows IIS server specified by the host parameter, described in the following table entry. <ul style="list-style-type: none"> <li>Any integer value is interpreted as the number of minutes.</li> <li>Integrity Client can not upload archived log files more frequently than the end-point computer is re-booted.</li> </ul> The enable element, described earlier in the preceding table entry, must be true to enable the frequency attribute.
<b>host</b> Specify the location of the Windows IIS Server and <i>IDLogUpload.asp</i> Active Server Page.	<b>Type:</b> Formatted string
	<b>Recognized values:</b> Valid pointer to the <i>IDLogUpload.asp</i> Active Server Page.
	Use the host attribute to specify the URL or IP address of the Windows IIS server to receive uploaded log files. For example:  <pre>http:\\101.102.103.104\inetpubs\wwwroot\UploadedLogs</pre> In the preceding example: <ul style="list-style-type: none"> <li>Either http or https (secure transfer mode) are valid transfer modes.</li> <li>The path name <code>\inetpubs\wwwroot\</code> is the Zone Labs-standard path name. If a different path name is specified, you must update the <i>RemoteRelativeDirectory</i> variable in the <i>IDLogUpload.asp</i> Active Server Page to reflect the new path.</li> <li>The last variable in the host attribute must be the name of the Active Server Page <i>IDLogUpload.asp</i>.</li> </ul> The enable element, described earlier in this table, must be true to enable the host attribute.

<i>/ZoneLabsSettings/configuration/autouploadlog (continued)</i>	
Attribute	Type, Values and Description <i>(Sheet 2 of 2)</i>
<b>retries</b> Specify how many times to attempt an upload.	<b>Type:</b> integer
	<b>Recognized values:</b> Integer number of retry attempts.
	Use the retries attribute to specify how many times Integrity Desktop will attempt to complete an unsuccessful upload. <ul style="list-style-type: none"> <li>The retries attribute operates in conjunction with the retryInterval attribute described in the following attribute table entry, and the timeout attribute, described later in this table.</li> <li>The enable element, described earlier in this table, must be true to enable the retries attribute.</li> </ul>
<b>retryInterval</b> Specify how long to wait between unsuccessful upload attempts.	<b>Type:</b> integer
	<b>Recognized values:</b> Integer number of seconds.
	Use the retryInterval attribute to specify how many seconds to wait between unsuccessful upload attempts. <ul style="list-style-type: none"> <li>The retryInterval attribute operates in conjunction with the retries attribute described in the preceding table entry, and the timeout attribute described later in this table.</li> <li>The enable element, described earlier in this table, must be true to enable the retryInterval attribute.</li> </ul>
<b>timeout</b> Specify how long to wait before abandoning an unsuccessful upload attempt.	<b>Type:</b> integer
	<b>Recognized values:</b> Integer number of seconds.
	Use the timeout attribute to specify how many seconds Integrity Desktop will wait before abandoning a download attempt. <ul style="list-style-type: none"> <li>The timeout attribute operates in conjunction with the retries attribute, and the retryInterval attribute, both described earlier in this table.</li> <li>The enable element, described earlier in this table, must be true to enable the timeout attribute.</li> </ul>
<b>Level 4 Child Elements</b>	
None	The <i>autouploadlog</i> element contains no child elements.

See also the XML Policy Reference for a complete description of the Zone labs XML Policy.

## Editing *autouploadlog* XML Policy Attributes

The *autouploadlog* element in a Zone Labs XML Policy has no corresponding Control Center section. Complete the following procedure to manually type the *autouploadlog* parameters into a configuration file.

### XML and Case-sensitivity

Unlike older style configuration files, XML is case sensitive: *autouploadlog* (the correct XML Policy capitalization) is not the same as *AutoUploadLog*, *autoUploadLog*, or any other variation of upper and lower case characters. Be sure to capitalize XML elements and attributes as written in the following procedure.

**To type *autouploadlog* parameters into an XML Policy file:**

- 1 Open the XML Policy file into a text-editing program. The default text-editing program for Windows is Notepad although you may prefer to use an editor designed to work with XML elements and attributes.
- 2 In the XML Policy file, locate the running ruleset's autoconfig element:
  - a Search for or browse to the ruleset that contains a name="runningruleset" attribute as shown in the following example:

```
<ruleset name="runningruleset" start="afterstartup" stop="onshutdown">
```

Only the running ruleset's autoconfig element will properly configure automatic upload or archived alert log data.
  - b From the running ruleset element, search for or browse forward (downward) in the XML Policy file for the autouploadlog child element. The following illustrates the general form of the autoconfig element (line break added):

```
<autouploadlog enabled="true" host="IIS_Server_Address/IDLogUpload.asp" frequency="boot"
  action="archive" timeout="60" retries="3" retryInterval="3"/>
```
- 3 In the XML Policy file, add or modify the autouploadlog element's attributes, including the quotation marks:
  - a Type enable="true"
  - b Type host="http://IIS\_Server\_Address/IDLogUpload.asp"
  - c Type frequency="IntegerValue", where *IntegerValue* specifies the number of minutes between uploads.
  - d Optionally, type a value for timeout, retries, and retryInterval.
- 4 Save the changes and exit the text-editing program.
- 5 After adding the autouploadlog parameters to the client computer's configuration file:
  - a Use an operational command line to direct Integrity Desktop to read the file.
  - b Restart Integrity Desktop to begin the alert log upload process. The next upload or archived alert logs occurs at the interval specified by the Frequency attribute.

After adding the autouploadlog parameters to the client computer's XML Policy file, use an operational command line to direct Integrity Desktop on the client computer to read the file.

**Editing the *autouploadlog* Section in a Configuration File**

A security policy contains a complete set of security and environment variables. In networks that do not have Zone Labs Integrity Server, use the autoupload section of a configuration file to schedule periodic uploads of archived event logs.

In the following example, the autouploadlog section of a configuration file specifies the download of a configuration file named *policy.ini*. Integrity Desktop downloads the file to, and reads the downloaded file from, the Integrity Desktop installation folder (the folder containing the program file *iclient.exe*).

In the following example, the `autouploadlog` section of a configuration file specifies path to the Active Server Page (ASP) file `IDLogUpload.asp`, as well as specifying the upload frequency, time, and retry intervals.

```
[autouploadlog]
Mode= On
DestinationURL=https://IIS_ServerAddress/IDLogUpload.asp
Frequency=Daily
Timeout=60
Retries=2
RetryInterval=120
```

The following section describes each of the `autoconfig` section's parameters in detail.

## List of *autouploadlog* Configuration File Parameters

Use the `autouploadlog` section of a configuration file on the client computer to enable automatic uploading of Integrity Desktop archived alert log files to a Windows IIS server.

The following table lists the seven parameters recognized by the `autouploadlog` section of a configuration file.

autouploadlog Parameters	Description
<b>DestinationURL=</b> <i>http://ServerAddress/IDLogUpload.asp</i> —or— <i>https://ServerAddress/IDLogUpload.asp</i>	URL or IP address of the destination server that will receive uploaded log files. <ul style="list-style-type: none"> <li>• <code>ServerAddress</code> is either the URL or IP Address of the computer running IIS.</li> <li>• The last variable in <code>DestinationURL</code> must be the name of the Active Server Page <code>IDLogUpload.asp</code>.</li> <li>• If a different path name is specified, update the <code>RemoteRelativeDirectory</code> variable in <code>IDLogUpload.asp</code> to reflect the new path.</li> </ul>
<b>Frequency=</b> { <i>Boot   Daily   Weekly   IntegerNumberOfMinutes</i> }	How frequently to upload the Integrity Desktop log file to the destination address. <ul style="list-style-type: none"> <li>• <code>Boot</code> uploads an alert log when Integrity Desktop (not the computer) is restarted.</li> <li>• Any integer value is interpreted as the number of minutes.</li> <li>• The smallest permissible interval is 60 minutes.</li> </ul>
<b>Mode=</b> { <i>On   Off</i> }	Enables or disables automatic uploading of log files.
<b>Retries=</b> <i>IntegerValue</i>	Number of times an unsuccessful log file upload will be retried.
<b>RetryInterval=</b> <i>IntegerNumberOfSeconds</i>	Number of seconds to wait before retrying an unsuccessful log file upload.
<b>Timeout=</b> <i>IntegerNumberOfSeconds</i>	Number of seconds before abandoning an unsuccessful log file upload.

## Editing *autouploadlog* Configuration File Parameters

The *autouploadlog* section of a configuration file has no corresponding Control Center section. Complete the following procedure to manually type the *autouploadlog* parameters into a configuration file.

### Before you Begin

You will need a previously created configuration file to complete the procedure in this section. If necessary, complete the procedure under “Saving Integrity Desktop Settings,” on page 44, before beginning the following procedure.

#### To manually type *autouploadlog* parameters into a configuration file:

- 1 Open the configuration file into a text-editing program. The default text-editing program for Windows is Notepad.
- 2 Perform one of the following:
  - In the Notepad window, locate the *autouploadlog* section of the configuration file  
—or—
  - If the configuration file does not contain an *autouploadlog* section, type `[autouploadlog]`
- 3 In the configuration file, below the *autouploadlog* section heading:
  - a Type `Mode=1`
  - b Type `DestinationURL=http://IIS_Server_Address/IDLogUpload.asp`
  - c Specify a value for Frequency.
  - d Specify a value for Timeout, Retries, and RetryInterval.
- 4 Save the changes and exit the text-editing program.
- 5 After adding the *autouploadlog* parameters to the client computer's configuration file:
  - a Use an operational command line to direct Integrity Desktop to read the file. See Chapter 5, “Using Operational Command Lines, for more information about operational command lines.
  - b Restart Integrity Desktop to begin the alert log upload process. The next upload or archived alert logs occurs at the interval specified by the Frequency parameter.

## The TempUploadResponseLogFile Receipt File

Each time log file connectivity is established the IIS server sends a receipt file named *TempUploadResponseLogFile.txt* to the client computer's `C:\%Windir%\System32\ZoneLabs` folder. (`%Windir%` is the Windows environment variable that points to a given computer's Windows root folder).

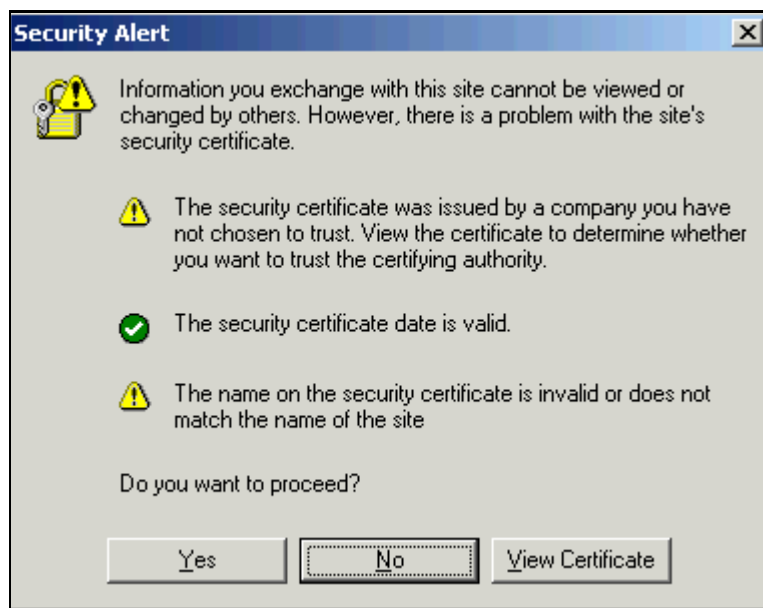
The following example shows the contents of a TempUploadResponseLogFile receipt file:

```
Uploaded directory : UploadedLogs/  
Uploaded file : 111.222.333.444SLAVEZERO_08-13-02_13-04-39.txt  
  
Remote IP Address : 111.222.333.444  
  
All Raw Headers: Host: 111.222.555.666  
User-Agent: ZoneLabs ZoneAlarm  
Pragma: no-cache  
Cookie: ASPSESSIONIDQGGQGUBU=DAGDEGKAIMPFDDLKJNIDHIAF  
Content-Length: 2181  
Content-Type: multipart/form-data80d8b7d7-77a0-4782-9210-27bed2ec703c  
  
Authorized User :  
Certificate Serial Number :  
Remote Host Name : 111.222.333.444  
Remote User Name :
```

Existing instances of TemUploadResponseLogFile are overwritten whenever a new receipt file is stored into the ZoneLabs folder.

## Responding to Certificate Alert Dialog Boxes

When using HTTPS to upload alert logs, each client computer's Web browser must be configured to accept the IIS server's certificate. Otherwise, when log uploads are attempted, a security alert similar to the one shown below appears on the client machine.



The Security Alert dialog box

- Click **Yes** to the Security Alert: the client computer will not generate additional Security Alerts during that session.
- To enable each client computer's browser to automatically accept your IIS certificate, refer to your system's on-line help.

## Locating and Viewing Log Files

Unless otherwise specified, Integrity Desktop stores log files to be uploaded in C:\Windows\Internet Logs.



**Empty alert log files are not uploaded:** This means that if no alerts have occurred during the period between scheduled uploads, Integrity Desktop does not perform alert log upload.

## Uploaded Log Files Naming Convention

Uploaded alert log files are stored in the Uploads folder as plain-text. The file name consists of the client computer name and date and time in twenty-four hour format.

The following example illustrates the general form of an uploaded log file name:

```
zapidlogtest_07_30_02_10_16_13.txt
```

In the preceding example:

- `zapidlogtest` is the name of the computer that uploaded the alert log file
- `07_30_02` indicates that the alert log file was uploaded on July 30, 2002
- `10_16_13` indicates that the alert log file was uploaded at 10:16:13 AM

## Example Log File Contents

The following lists the contents of a typical log file (line breaks added for readability).

```
ZoneAlarm Logging Client v4.0.059
Windows NT-5.0.2195-Service Pack 2-SP
type,date,time,source,destination,transport
FWIN,2002/04/30,12:48:37 -7:00 GMT,216.236.205.198:2755,216.15.66.231:445,TCP (flags:S)
FWIN,2002/04/30,12:48:37 -7:00 GMT,216.236.205.198:2756,216.15.66.231:139,
TCP (flags:S)
FWIN,2002/04/30,12:48:37 -7:00 GMT,192.168.0.1:2757,216.15.66.231:139,
TCP (flags:S)
PE,2002/04/30,22:27:21 -7:00 GMT,Generic Host Process for Win32 Services,
0.0.0.0:N/A
```

## Reading Uploaded Alerts

The following example illustrates the general form of an alert log entry.

Type,Date,Time,Source,Destination,Transport

The following table list each of the six fields in the order they appear in a file entry.

Field	Description
<b>Type</b>	The type of event. See the table under “Event Types,” in the following section, for a detailed description of each of the eight Integrity Desktop event types.
<b>Date</b>	The date of the alert as <i>yyy/mm/dd</i> .
<b>Time</b>	The local time of the alert. This field also displays the hours difference between the computer’s local and Greenwich Mean Time (GMT) time zones.
<b>Source</b>	Can be one of two values: <ul style="list-style-type: none"> <li>• The IP address of the computer that sent the blocked packet and the port used.</li> <li>—or—</li> <li>• The program on the client computer that requested access permission.</li> </ul>
<b>Destination</b>	The destination IP address and port of a blocked packet.
<b>Transport</b>	The data transfer protocol of the alert.

## Event Types

The following table describes each of the eight Integrity Desktop event types.

Event Type	Description
<b>ACCESS</b>	<p>The ACCESS entry indicates that Program Control prevented a program from accessing remote resources. The following example illustrates an ACCESS event entry (line breaks added for readability):</p> <pre>ACCESS, 2000/09/07,16:45:57 -5:00 GMT, Microsoft Internet Explorer was not allowed to connect to the Internet (64.55.37.186)., N/A,N/A</pre> <p>The preceding example illustrates the placement of the following ACCESS fields:</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• The program that attempted to access the Internet</li> <li>• The IP Address the program was trying to connect to</li> </ul>
<b>FWIN</b>	<p>FWIN (firewall, inbound) indicates that the firewall blocked an incoming request; some, but not all, FWIN events are connection attempts. The following example illustrates an FWIN event entry (line breaks added for readability):</p> <pre>FWIN, 2000/03/07,14:44:58,-8:00 GMT, Src=192.168.168.116:0, Dest=192.168.168.113:0, Incoming, ICMP</pre> <p>The preceding example illustrates the placement of the following FWIN fields:</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• Source IP Address and port number</li> <li>• Destination IP Address and port number</li> <li>• The protocol as either TCP, UDP, ICMP, or IGMP</li> </ul> <p>When ZoneAlarm Pro blocks an Internet Control Message Protocol (ICMP) packet or TCP packet, the log entry also includes supplemental identifiers.</p> <ul style="list-style-type: none"> <li>• ICMP message identifiers are listed under “ICMP Message Types,” on page 88.</li> <li>• ICMP packet type identifiers are listed under “TCP Packet Type Flags,” on page 89.</li> </ul>
<b>FWLOOP</b>	<p>FWLOOP (firewall loopback) indicates that the firewall blocked a loopback packet addressed to the Network Interface Card (NIC). The address of system loopbacks is generally 127.0.0.1.</p>

Event Type	Description <i>(continued)</i>
<b>FWOUT</b>	<p>FWOUT (firewall, outbound) indicates that the firewall blocked an outbound request from a program. The following example illustrates an FWOUT event entry (line breaks added for readability):</p> <pre>FWOUT, 2000/03/07,14:47:02,-8:00 GMT, QuickTime Player Application tried to access the Internet. Remote host: 192:168:1:10</pre> <p>The preceding example illustrates the placement of the following FWOUT fields:</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• The program that attempted to access the Internet</li> <li>• The remote host the program was trying to connect to</li> </ul>
<b>FWROUTE</b>	<p>FWROUTE (firewall, routing) indicates that the firewall blocked a packet that was not addressed to or from the computer, but that attempted to route through it.</p>
<b>LOCK</b>	<p>LOCK (Integrity Desktop Internet Lock) indicates that a program attempted to access the Internet while the Internet Lock was engaged. The following example illustrates a LOCK event entry (line breaks added for readability):</p> <pre>LOCK, 2000/09/07,16:43:30 -7:00 GMT, Yahoo! Messenger, 207.181.192.252, N/A</pre> <p>The preceding example illustrates the placement of the following LOCK fields:</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• The program that attempted to access the Internet</li> <li>• The IP Address the program was trying to connect to</li> </ul>

Event Type	Description <i>(continued)</i>
<b>MS</b>	<p>MS (Mailsafe) events indicates that MailSafe quarantined a file received by your e-mail client. The following example illustrates a MS event entry (line breaks added for readability):</p> <p>MS, 2000/09/08,09:45:56 -5:00 GMT, Microsoft Windows(TM) Messaging Subsystem Spooler, Renamed e-mail attachment of type .HLP to .zla,N/A</p> <p>The preceding example illustrates the placement of the following MS fields:</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• The e-mail delivery mechanism; in the preceding example it is Microsoft Windows(TM) Messaging Subsystem Spooler</li> <li>• The name and type of the file that was quarantined.</li> </ul>
<b>PE</b>	<p>PE (Program Event) indicates that a program attempts to access the Internet. The following example illustrates a PE event entry (line breaks added for readability):</p> <p>PE, 2000/03/22,17:17:11 -8:00 GMT, Netscape Navigator application file,192.168.1.10</p> <ul style="list-style-type: none"> <li>• Date and time</li> <li>• The program that attempted to access the Internet</li> <li>• The IP Address and Port number the program was trying to connect to</li> </ul>

## ICMP Message Types

When ZoneAlarm Pro blocks an inbound Internet Control Message Protocol (ICMP) packet, the log entry also identifies the type of ICMP message.

Code	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request

Code	Description <i>(continued)</i>
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

## TCP Packet Type Flags

TCP Flags identify specific TCP packet types.

Flag	Description
4	Low-order unused bit
8	High-order unused bit
A	ACK
F	FIN
P	Push
R	Reset
S	SYN
U	URGENT

---

## Supplemental Administrator Utilities

---

Zone Labs, Inc. has created two supplemental utilities to aid in the detection and reconfiguration of Zone Labs products:

- A Product Finder Utility, described in the following section, scans the client computer for previously installed instances of Zone Labs products.
- A Product Update Utility, describes on page 91, simplifies and enhances the update of Integrity Client security and configuration settings.

Contact your Zone Labs support representative to obtain either of these free utilities.

### Running the Product Finder Utility

The Zone Labs Product Finder Utility *productfind.exe* finds any Zone Labs products previously installed on a computer, and outputs the results to a text file named *check.log*.

#### To run the Product Finder Utility:

- 1 Copy the Product Finder Utility program file *productfind.exe* to a Windows folder.
- 2 Perform one of the following:
  - Double-click the Product Finder Utility program's icon
  - From a DOS command line, type `Productfind.exe /S`

The Product Finder Utility:

- Examines the Windows registry.
- Creates a file named *check.log* in the same Windows folder that contains the Product Finder Utility.

The *check.log* file produced by the Product Finder Utility contains a single-digit code that corresponds to a particular Zone Labs product. The following table lists the single-digit codes output by the Product Finder Utility and the Zone Labs product that corresponds to that single-digit code.

Code	Product Name
0	No Zone Labs product installed
1	Integrity Client, including Integrity Agent version 3.1 and higher
2	ZoneAlarm Pro
3	ZoneAlarm Plus
4	ZoneAlarm

Code	Product Name <i>(continued)</i>
5	Integrity Agent, versions 3.0 or lower
6	Zone Alarm Pro - Integrity Desktop
9	Unknown product or Zone Labs OEM product

## Using the Policy Update Utility

Use the Policy Update Utility to perform simple reconfiguration of Integrity Client.

### Policy Update Utility File-naming Conventions

The Policy Update Utility program file *update.exe* operates in conjunction with three supplemental configuration files:

- *config.xml* contains security and configuration settings in Zone Labs XML Policy format. See the Integrity *Client Reference Guide* for a complete description of XML Policy elements and attributes.
- *update.ini* contains an optional upgrade key value as well as optional startup or shutdown messages. See “Specifying Policy Update Parameters,” in the following section, for a list of *update.ini* parameters and values.
- *registry.reg* contains Windows Registry keys and values. See “Specifying Windows Registry Keys and Values,” on page 92, for an overview of the *registry.reg* file.

XML Policy files can be created using Integrity Server’s Policy Studio or the Integrity Client Control Center (graphical user interface).

You can use a simple text editor, such as Notepad, to create or edit each of these files.

## Specifying Policy Update Parameters

The Policy Update Utility can read settings as well as startup or shutdown messages from an *update.ini* file. The following table lists the *update.ini* parameters and values recognized by the Policy Update Utility.

Parameter	Function
<b>Close</b> = <i>Free-form String</i>	Use the Close update file parameter to have the Policy Update Utility display a closing message immediately before the update utility stops.  If no text is specified, the closing message is not shown. See also the Open parameter, described later in this table.
<b>Key</b> = <i>UpgradeKeyOld</i>	Use the Key update file parameter to have the Policy Update Utility perform a silent reconfiguration.  An upgrade key must have been specified at the time Integrity Client was installed.  See also “/upgradekey,” on page 26.
<b>Open</b> = <i>Free-form String</i>	Use the Open update file parameter to have the Policy Update Utility display an opening message immediately after the update utility starts.  If no text is specified, the opening message is not shown. See also the Close parameter, described earlier in this table.
<b>suppressreboot</b> =[ <i>Yes   No</i> ]	Specify the suppressreboot update file parameter equal to Yes to have the Policy Update utility suppress the reboot of the computer after reconfiguration is complete.  Suppressing reboot merely defers the reboot to a more convenient time, such as until after a multi-task upgrade script has completed.

## Specifying Windows Registry Keys and Values

The Policy Update Utility can read new or modified values for the Windows Registry from a *registry.reg* file.



**Modifying keys and values in the Windows Registry must be performed with extreme care. Incorrectly specified Registry values may prevent the proper operation of Windows or any programs running on the computer.**

The Windows Registry contains *keys* and *values* that specify a wide range of Windows operating system settings, file locations, and operational behaviors. To acquire more information about the Windows Registry, including Registry keys and values, consult the following information sources:

- In the Windows on-line help, search for the words *Windows Registry* or *regedit*.
- In the Windows Registry Editor utility (regedit), browse the on-line help

## Running the Policy Update Utility

Complete the following procedure to run the Policy Update Utility.

### To use the Policy Update Utility to reconfigure Integrity Client:

- 1 Place the Policy Update Utility program file *update.exe* anywhere on the client computer.
- 2 In the same folder as *update.exe*, place:
  - A policy file named *config.xml*.
  - An optional update file named *update.ini* containing parameters as described under "Specifying Policy Update Parameters," on page 92.
  - An optional registry file named *registry.reg* containing new or modified Windows Registry keys and values.
- 3 Run the Policy Update Utility. Perform one of the following:
  - In the folder containing the *update.exe* program file, double-click on the file to start the Policy Update Utility
  - In a DOS command window, browse to the folder containing the *update.exe* program file and type *update.exe*

The Policy Update Utility reconfigures Integrity Client with the policy settings contained in the supplemental files described under "Policy Update Utility File-naming Conventions," on page 91.

## Managing Passwords and Files

Both the Integrity Client installer and Integrity Client programs exchange and store certain information in clear-text form. Though not generally a problem, this appendix suggests the following two ways to protect clear-text information and files:

- ““Managing Clear-text Passwords,” in the following section
- “Managing Clear-text Data Files,” on page 96

### Managing Clear-text Passwords

During installation, the Integrity Desktop installation program prompts for an installation-level password. An installation-level password typed in response to this prompt is submitted as clear-text: the password is neither encrypted nor concealed from an onlooker or a program such as a key-logging Trojan horse.

This section describes two approaches to managing clear-text Integrity Desktop passwords:

- ““Using Windows Script Files,” in the following section
- “Using Compiled Programs,” on page 95

### Using Windows Script Files

One way to conceal a plain-text password is to create an installation script. A Windows batch file or script file (.bat, .cmd, or .vbs file name extension) runs the Integrity Desktop installation program and passes it all the command line switches, including the password.

The following table lists the advantages and disadvantages of using a script file to manage a clear-text password.

Advantages of Script Files
<ul style="list-style-type: none"><li>• A script helps conceal the password from novice users and malicious key-logging programs.</li></ul>
Disadvantages of Script Files
<ul style="list-style-type: none"><li>• A knowledgeable user can read the plain-text contents of a script so access to the script must be managed.</li><li>• When a batch or script file (.bat or .cmd file name extension) is executed, commands can be observed as they are passed to the command window.</li><li>• Earlier versions of Windows may require downloading and installation of Microsoft's Windows Scripting Host (WSH). Some security departments disable the Microsoft Windows Scripting Host (WSH) necessary to run Visual Basic Scripts.</li></ul>

In summary, script files provide an easily implemented way to conceal plain-text passwords from casual observation. Script files do not, on the other hand, deter a moderately knowledgeable user from reading the contents of the plain-text script.

## Tips on Using Script Files

Some simple ways to enhance the basic password protection provided by a script file are:

- Keep the script file on a central server rather than multiple workstations. (See also “Managing Custom Configuration Files,” on page 27).
- If the script file is distributed: Include two lines in file: one to delete the script file, and one to clean the recycle bin so the deleted script can't be easily recovered.
- Set the files read, write, and execute permissions to make casual reading more difficult.

## Using Compiled Programs

A second, more effective, way to conceal plain-text passwords is to compile the install commands into a program.

The following table lists the advantages and disadvantages of using a compiled program to manage a clear-text password.

<b>Advantages of Compiled Programs</b>
<ul style="list-style-type: none"> <li>• The Integrity Agent installer is invoked silently and the password is effectively concealed from onlookers.</li> <li>• Any programming language can be used.</li> </ul>
<b>Disadvantages of Compiled Programs</b>
<ul style="list-style-type: none"> <li>• Requires knowledge of writing and compiling a program.</li> <li>• Decompiling the program is still possible, so access to the compiled program must be managed.</li> </ul>

## Tips on Using Compiled Programs

Some simple ways to enhance the password protection provided by a compiled program are:

- Use compiled Visual Basic Script (“.vbs” file name extension). VBS is very similar to Visual Basic and is a less technically demanding way to create a compiled installation script.
- Keep the compiled installation program on a central server rather than multiple workstation's. See also “Managing Custom Configuration Files,” on page 27).
- If the file is distributed, include two commands in the program: one to delete the program file, and one to clean the recycle bin so the deleted program can't be easily recovered.

## Managing Custom Configuration Files

In addition to managing clear-text passwords, the accidental release of the configuration file might expose details of the corporate security policies and the IP address of the Integrity Server for that client.

Zone Labs, Inc. therefore recommends:

- Centrally storing and controlling access to any optional configuration files

- Deleting the files after installation is completed

After installation, the contents of the configuration file are merged into the Integrity Agent and thereafter maintained in an encrypted state.

## Managing Clear-text Data Files

The Integrity Desktop programs use clear-text files to store two types of information:

- Policy file information, containing IP addresses, known programs and components, and other configuration information
- Alerts and log files that may contain information of value to system intruders

The following sections describe ways to minimize unauthorized access both types of files.

## Managing Clear-text Policy Files

Policy files (default name `policy.ini`) contain information of potential use to a system intruder. To make this information more difficult to Zone Labs, Inc. recommends that policy files be stored only on a central server or on an Integrity Server.

## Managing Clear-text Alerts and Log Files

The Integrity Desktop client programs store alerts and log files in the `C:\%windir%\Internet Logs\` folder, where `%windir%` is a Windows system variable that identifies where Windows program files are stored.

In some cases, alerts and log files may contain information—such as the names of programs running on a system—that may be useful to intruders. Because of this it is recommended that you upload then delete alerts and log files on a regular basis.

## 0-9

3DES Data Encryption Standard 48

## A

Active Server Page 70

Advanced Alerts and Log Settings 76

alert log uploading

    default file names 74

    destination folder for 73

    destination folder permissions 73

    enabling 74

    folder hierarchy 74

    Log Control tab 76

    overview of tasks 72

    protocols supported 71

    requirements for 70

alert logs

    as troubleshooting aid 58

    settings dialog box 59

alerts and logs files

    managing 96

    protecting contents of 96

anti-virus, Integrity Desktop and 1

AutoCheck, values for 67

AutoConfig 63–66

autoconfig, as XML Policy element 63

autouploadlog

    as XML Policy element 77

    parameters 81–82

    specifying 82

## B

BlockFragments, and VPN connections 56

bold, use of 5

braces, use of 5

brackets, use of 5

## C

case sensitivity, in XML Policy 65, 79

cautions 6

certificates

    alert dialog boxes 83

    Security Alert dialog box 83

check.log 90

clean switch

    availability of 47

    deprecated 47

clean uninstallation, as default 47

codes, product 90

command lines, see installation

    command lines or operational

    command lines

command lines, types of 14, 35

config command line switch

    and Policy\_Info section 16, 41

    general form of 41

    preceding by dash 14, 35

    syntactic requirements 14, 16, 35

configuration files

    "pulled" 62

    and Integrity Desktop 4.0 62, 70

    and policy switch 33

    and slash mark 33

    AutoConfig 63–66

    compatibility with 62, 70

    continued support for 62, 70

    file and pathname specifier 14, 35

    general form of 33

    in Integrity Agent or Flex 44

    policy\_Info section ignored in 33, 41

    post-installation use of 16, 36

    protecting 95

    requirements for downloading 62

    specifying during installation 33

    using 62

    viewing 44

Configuration Wizard, see wizard

consumer products, upgrading 10

Control Center

    availability of 4

    displaying after installation 21

CTRL-ALT-double-click 44

## D

dash

    use of 28, 36

data files, protecting 96

database files

    Integrity Client 70

    Zone Alarm 11

default installation 14

default operating modes 4

DES (Data Encryption Standard) 48

dialog box

    Advanced Alerts and Log Settings 76

    Security Alert 83

document naming conventions 4

downgrading

    and user settings 10

    rules for 10

## E

ellipsis, use of 5

enterprise security policies 3

    "pulled" 62, 69

errlog switch

    and s switch 18

    default value for 18

    general form of 18

error log file, location of 18, 25

error message, command line 16

ESP (Encapsulating Security Payload protocol) 48

event types, log file 86–88

## F

file naming conventions, installation programs 4

fragmented packets, see packet

    fragments

fragments, see packet fragments 56

## G

GRE (Generic Routing Encapsulation protocol) 48

## H

H.323 49

hardware, required 8

Host/Site, defining 52

hyphens, in long command lines 5, 6

## I

i switch

    default value for 28

    general form of 28

iclientSetup\_IAen.exe 4

iclientSetup\_IDen.exe 4, 13

iclientSetup\_IFen.exe 4

IDLogUpload.asp 70, 72

IDSetup\_110n.exe 14, 35

IIS issues 73

IIS, see Microsoft

IIS, see Microsoft IIS

IKE (Internet Key Exchange protocol) 49

install\_log

    conditions for use 16

    location of 16

install\_log switch

    default value for 19

    general form of 19

installation

    default behavior of 14

    display of wizard during

    performing default 13

installation command lines  
  compared with operational command lines 14, 35  
  delimiters in 14, 15, 18, 35  
  elements of 15  
  error messages in 16  
  general form of 15  
  limitations on size 15  
  overview 14, 35  
  overview of differences between 14, 35  
  switches in 17–33  
  used for 17

installation deployment tools 34

installation log file, location of 19

installation program 4, 13

installation-level password  
  compared to user-level password 28  
  reset of 31, 40  
  scope of 28

installldir switch  
  and invalid path and file names 19  
  and quotation marks 19  
  and s switch 19, 25  
  default value for 19  
  general form of 19

Integrity Client  
  definition of 4

Integrity Desktop  
  alert log uploading 3  
  and Integrity Server 13  
  and Web servers 3  
  installation program 13  
  network types 3  
  policy arbitration 4

IP addresses  
  defining to Integrity Client 52

IP Range, defining to Integrity Client 52

IPSec (IP Security Protocol) and Windows 49

italic, use of 5

**L**

L2F (Layer 2 Forwarding protocol) 49

LDAP  
  see Lightweight Directory Access Protocol

License key  
  installation command line switch 37

license key  
  format for 37  
  installation command line switch 16

license key, see lickey switch

lickey switch  
  default value for 18, 20, 22, 37  
  general form of 18, 20, 37

lickey, see License key

Lightweight Directory Access Protocol 49

line breaks, in command lines 5, 6

log files  
  contents of 84  
  Date parameter 85  
  Destination parameter 85  
  event types 86–88  
  location of 84  
  naming conventions for 84  
  packet type flags 89  
  reading 85  
  Source parameter 85  
  Time parameter 85  
  Transport parameter 85  
  Type parameter 85  
  viewing 84

long command lines 5

## M

malware 1

Microsoft  
  IIS 63, 71, 72, 82  
  Manual of Style 5  
  security issues 72  
  supported operating systems 8  
  Systems Management Server 20, 22, 34  
  Universal Naming Convention 18

## N

networks  
  defining to Integrity Client 52

noreboot switch  
  and SMS 20  
  default value of 20  
  general form of 20  
  installation versus upgrade 20  
  required by upgrade 20

nostartup switch  
  default value for 21, 24  
  general form of 21, 24

notes 6

notutorial switch  
  default value for 27  
  general form of 27

Nowizards switch  
  default value for 28  
  general form of 28

## O

operating systems, supported 8

operational command lines  
  compared with Installation command lines 14, 35  
  delimiters in 14, 35, 36  
  elements of 36  
  overview 14, 35

overview panel  
  Status tab 44

## P

packet fragments  
  and BlockFragments parameter 56  
  and VPN connections 56

parameter values  
  equivalency 6

passwinstset switch  
  default value for 32  
  general form of 32

password switch  
  and passwset switch 29  
  default value for 29, 39  
  general form of 29, 39

passwords  
  and compiled programs 95  
  and Trojan horse 94  
  behaviors of 28  
  modifying 40  
  protecting 94  
  recommendations for 38  
  scope of 28  
  setting 38  
  syntactic requirements for 38, 40  
  threats to 94

passwset  
  default value for 38

passwset switch  
  default value for 29  
  general form of 29, 38  
  syntactic requirements for 29  
  syntactic requirements of 31

path and file names, enclosing in quotation marks 14, 35

Pentium, versions supported 8

pipe symbol, use of 5

Policies panel  
  availability of 4

policy arbitration 4

policy files  
  managing 96  
  protecting contents of 96  
  types of 34

Policy switch, incompatibility with Integrity Desktop 41

Policy Update  
  parameters 92  
  supplemental files 91, 93

policy\_Info section  
  and config switch 41

policy\_Info section, ignored by config command line switch 16

port  
  scanning of 1

PPTP  
  Point-to-Point Tunneling Protocol 49

product codes, output by Product Finder 90

Product Finder  
  check.log 90  
  output file 90  
  product codes 90  
  productfind.exe 90

programs  
  granting access to 53  
  hierarchy of permissions 55  
  trusted 2

pwinst switch  
  and pwinsset switch 32  
  default value for 32, 40  
  general form of 32, 40

pwinstset switch  
  and pwinst switch 32, 40  
  and reset switch 32  
  default value for 40  
  general form of 40

**Q**

quotation marks  
  use of 14, 33, 35, 41

**R**

RADIUS, see Remote Authentication  
  Dial-In User Service

RAM, minimum requirement 8

RAS, see H.323

rbprompt switch  
  and GUI reboot prompt 22  
  and s switch 22  
  default value for 22  
  general form of 22, 23

reboot  
  after upgrade 20  
  and initial installation 10  
  and s switch 25  
  messages 22  
  required after upgrade 10

reboot, forcing after installation 22

receipt files 82

registry, Windows 92

registry.reg 91

Remote Authentication Dial-In User  
  Service, VPN support for 49

required hardware 8

required software 8

reset  
  and user preferences 11

Reset switch  
  and pwinstset switch 32

reset switch  
  use of 17

reset switch, scope of 24

reset switch, use of 24

running ruleset, configuration of 66, 80

**S**

s switch  
  and errlog switch 18  
  and error log 25  
  and installdir switch 19, 25  
  and rbprompt switch 22  
  and reboot 25  
  default value for 25  
  general form of 25  
  position of 24  
  risks of using 25  
  syntactic requirements 24  
  used during upgrade  
  used with errlog switch 18

script files  
  for passwords 94  
  pros and cons 94  
  suggestions for use 95

security  
  review of practices 73

Security Alert dialog box 83

security policies  
  "pulled" 3  
  "pushed" 3

security, Microsoft IIS 72

servers, access permissions for 55

setup Wizard 14

silent upgrade, see s switch

SKIP (Simple Key Management for  
  Internet Protocol) 49

slash mark  
  use of 15, 28, 33, 41

SMS  
  see Microsoft, Systems Management  
  Server

software, required 8

Status tab, Overview panel 44

subnet  
  defining to Integrity Client 53

Syntactic conventions  
  slash mark 33

syntactic conventions  
  bold 5  
  braces 5  
  brackets 5  
  dash 36  
  ellipsis 5  
  hyphen 5, 6  
  in XML Policy 65, 79  
  italic 5  
  slash mark 15, 41

system requirements 8

**T**

TempUploadResponseLogFile 82  
  example of 83

tips, see notes

Trojan horse 1, 2

Trojan horse, and passwords 94

troubleshooting  
  removing temporary settings for 60  
  settings for 60  
  techniques for 57  
  using Program Learning mode for 60

tutorial  
  as part of installation 14

tutorial, controlling display of 27

**U**

UNC see Microsoft Universal Naming  
  Convention

UNC, see Microsoft Universal Naming  
  Convention

uninstalling  
  and install\_log switch 46  
  and user settings 9  
  clean 46  
  default 46  
  required programs 46  
  Zauninst.exe 46

update.exe 91

update.ini 91

upgrading  
  and consumer products 10  
  and reboot switch 10  
  and user settings 9, 12, 18  
  completing 20  
  guidelines for 9  
  rebooting after 20  
  requirements for 9  
  silent 22

uploaded log files, see log files

uploading, alert logs 70

user settings 9

user-level password  
  compared to installation-level  
  password 28  
  recommendations for 28, 29, 38  
  reset of 32  
  scope of 28

**V**

VBS, see Visual Basic Script  
  viewing configuration settings 44

Virtual Private Network  
  allowing protocols for 57  
  Cisco 3  
  concentrators or gateways 50  
  granting network access in 53  
  optional resources 50–51  
  overview of steps 49  
  required network resources 50  
  responding to Alert boxes for 54  
  supported protocols  
  troubleshooting of 57

Visual Basic Script  
  and passwords  
  performing cleanup with

VPN, see Virtual Private Network

---

## W

WAN (wide-area Network) 50

warnings, see cautions

wide-area network 50

Windows

- and installation command lines 15

- and VPN support 48

- command line limitations 15

- IIS

- registry keys and values 91, 92

- Scripting Host (WSH) 94

- System Tray icon 44

- Universal Naming Convention 33, 41

WinZip 34

Wizard

- setup 14

WSH, see Windows Scripting Host

## X

XAUTH (Extended Authentication protocol) 49

XML Policy

- and autoconfig element 63

- and autouploadlog element 77

- and Integrity Desktop 4.0 62, 70

- autoconfig section 63

- case sensitivity in 65, 79

- general format of 63, 77

## Z

Zauninst.exe

- location of 46

- obtaining 46