

Triage User Manual

Copyright © MetaQuest Software Inc. All rights reserved.

The information in this document is subject to change without notice and should not be construed As a commitment by MetaQuest Software Inc. MetaQuest Software Inc. assumes no responsibility for errors in this document.

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

Metaquest and Triage are trademarks of Metaquest Software Inc. All other trademarks are the property of their respective owners.

Contents

Chapter 1	
Getting Started	1
The Triage eSupport Process.....	1
Triage Components	1
About Triage Console.....	2
Installing Triage.....	3
QuickStart.....	4
Chapter 2	
Building Profiles	7
Profiling and the Windows Operating System.....	7
Building Profiles	7
Auto-profiling Applications	8
Importing Windows Installer (MSI) Packages.....	8
MSI Files and Self-Registered DLLs.....	9
Importing Install Packages	9
Installed Applications	10
After You Auto Profile	10
Manually Building a Profile	11
Chapter 3	
Performing Change Analysis	13
Creating a Reference Audit Report.....	13
Adding Audit Reports to Triage Console.....	13
Viewing Audit Reports.....	13
Comparing Audit Reports	14
Filtering Audit Reports	15
Customizing the Difference Highlighting.....	16
Hiding Files from Non-Active Operating Systems.....	16
Printing Audit and Diagnostic Reports	16

Chapter 4	
Collecting Diagnostics and Configuration Information	17
Defining Variables.	17
Using Regular Expressions.	23
System Resources	24
Auditing Files	25
Auditing ActiveX Controls	26
Auditing Registry Keys and Entries.	26
Auditing Shortcuts.	27
Copying Files	27
Auditing with Windows Management Instrumentation	28
Auditing Database Information	30
Collecting Diagnostics for IIS	32
Collecting Security Information	32
Collecting User Information.	32
Chapter 5	
Adding Self-Repair Capabilities	33
Setting Up Self-Repair	33
Generating Repair Rules	34
Editing Repair Rules	34
Defining Conditions.	34
Defining Actions	35
Setting Attribute Values	36
Enabling and Disabling Repair Rules	37
Locking Customized Repair Rules	38
Creating Self-Repair Packages.	38
Configuring Self-Repair.	38
Chapter 6	
Configuring Triage Client	41
Interface Modes	41
Configuring Classic Mode	41
Configuring Wizard Mode	42
Command-Line Arguments	44
Web Publishing Audit Reports	46

Chapter 7	
Setting Up Triage eSupport	49
Signing	49
Customizing eSupport Web Pages	50
Generating eSupport Packages	51
Specifying the Triage Client Command Line	52
Chapter 8	
Distributing Triage Client on CD	53
Chapter 9	
Configuring	55
Moving the Support Site	55
The Support Site User Account	55
Event Logging	56
Licensing	56
Audit Preferences	57
Audit Preferences	57
Client Preferences	58
eSupport Preferences	58
Mail Preferences	59
Security Preferences	59
Self Repair Preferences	59
Product Preferences	60

Chapter 1: Getting Started

Triage provides automated eSupport for any type of software application or hardware peripheral. With Triage, you can automate the collection of accurate diagnostics and problem resolution, as well as speed up root cause analysis.

The Triage eSupport Process

- 1 Use Triage Console to build an application profile that specifies exactly what diagnostics you want to collect. Then generate an eSupport page that you can add to your technical support Web site.
- 2 When users encounter a technical problem, they go to the eSupport page on your Web site. From this eSupport page, users can download a self-extracting eSupport package that contains Triage Client and an application profile. This eSupport package automatically runs Triage Client, which loads the application profile. This is all transparent to the user, who simply has to click a button.

Triage Client can also be packaged for distribution on CD.

- 3 Triage Client automatically gathers the specified diagnostics, along with information about the user and a description of the problem.

Triage Client notifies users of any potential problems found during the diagnostic audit, and allows them to apply automated self-repairs.

- 4 The collected diagnostics are stored in a digital audit report, which is transferred over the Internet back to your technical support department.
- 5 To diagnose problems, you use Triage Console to visually compare audit reports from user systems against audit reports for working systems, and quickly focus in on possible root causes.

Triage Components

Triage consists of:

- Triage Console, a central, administrative console.
- Triage Client, a downloadable agent that runs on end-user computers.
- Support Site, a data folder shared by all instances of Triage Console.

Triage Console

You use the Triage Console to build application profiles that specify what diagnostics to collect, generate eSupport Web pages and CD distribution packages, and analyze collected diagnostics.

The Triage Console is a Microsoft Management Console (MMC) snap-in that you can start from the PC-Duo Enterprise Console or run as a standalone application. MMC is a feature of the Windows 2000, NT, and XP operating systems, but can also run on the Windows 95, 98, and Me operating systems. You can add the Triage snap-in to other MMC consoles.

Triage Client

A configurable eSupport agent program that can be downloaded from a Web page. Triage Client automatically gathers accurate configuration and problem diagnostics from end-user systems, and can automatically repair potential problems. (Self-repair requires Triage Pro.)

Support Site

Support Site is a shared folder that has the following functionality:

- Stores the public profiles and audit reports, and the licensing information.
- It includes the setup program for additional Triage consoles. After the first Triage Console is installed and configured, all other copies of the console are installed from Support Site.

About Triage Console

The Triage Console consists of a window divided into two panes. The left pane contains the console tree, which shows the items available in the console.

The right pane contains the Details view. The Details view shows information about the item selected in the console tree. For example, when you click a profile in the console tree, the Details view allows you to view and edit the details of the profile.

Action Menu

Most tasks in Triage Console, such as profiling applications and running audits, can be accomplished from the **Action** menu.

The available commands on the Action menu depend on what type of item you select in the console tree. Right-clicking an item in the console tree opens a shortcut menu with the same commands.

Console Tree

From the console tree, you can create and edit profiles, review diagnostics, and analyze configuration changes.

Profiles A profile specifies what configuration information to collect. In addition to files, registry entries, ActiveX controls, shortcuts, and environment variables, a profile can include lists of files to retrieve and system resource information (such as services, startup applications, and printers).

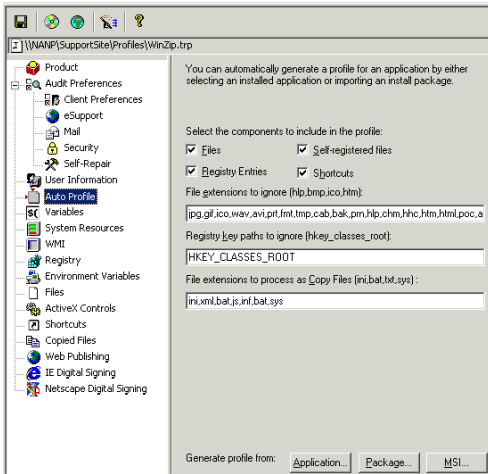
Public versus Private Public items are stored on a central server (in the SupportSite shared folder) and shared by all console users. For example, if you create a new profile you can share it with all other users by saving it in the SupportSite folder. Private items are stored outside of the SupportSite folder, for example on your local hard disk.

Audits An audit is the configuration information and diagnostic data collected from a computer. Audit reports can be stored in the Support Site or anywhere on your network.

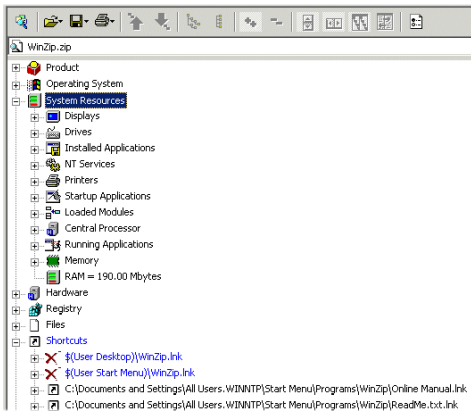
Details View

The Details view displays the details of an item selected in the console tree. For example, you can view the details of a profile, an audit report, or of the problems found during an audit.

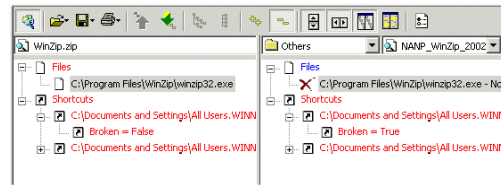
Profile View Allows you to create and edit profiles. The left pane of the profile view is the profile tree, which lists the different sections of a profile.



Audit Report View Allows you to review the contents of an audit report.



Change Analysis View Allows you to compare two audit reports. Differences between the two reports are visually highlighted, so you can quickly view problems such as missing files, wrong file versions, invalid registry entries, and invalid OS settings.



Installing Triage

You can install a copy of Triage from the CD. After you install Triage, the Setup program allows you to set options that will be shared by all consoles (such as the Support Site location).

By default, the Setup program installs all features in the default location. To select the features to install, choose the Custom setup type.

Select Features The Custom setup type allows you to select the features to install. You must install Triage. Install Triage Pro if you want to use self-repair or WMI. If you don't want to diagnose ODBC database problems, you don't need to install Triage/db.

Setting Up the Support Site The Support Site is a shared folder on a network server. All consoles must be able to access the Support Site.

Setting the Support Site User Account The Support Site user account is used by all consoles to access the Support Site shared folder.

The Support Site user account must have the appropriate privileges on each local computer to perform tasks such as saving audit reports. Ideally,

the Support Site user should be a Domain Administrator that has local Administrative privileges on each computer.

Setting the Event Logging Options By default, consoles log events on the local computer. On Windows NT, 2000, and XP, events are logged to the Event Log. On Windows 95, 98, and Me, events are logged to a text file.

On Windows NT, 2000, and XP, you can log all events to the Event Log on a central server. See “Event Logging” on page 56 for more information.

Registering Triage Each Triage product (Triage Console, Diagnostics Agent, Triage/db) must be registered with its serial number. See “Registering Products” on page 56 for more information.

Installing Additional Consoles

You can install any number of additional consoles (each copy requires a separate license).

To install additional Triage Consoles:

- 1 Connect to the Support Site shared folder.
- 2 Run the setup program in Setup\Console.

QuickStart

This quick start shows you how to use Triage to automate the collection of diagnostic data for an application.

At the end of this quick start lesson, you will have an HTML page that you can add to your company web site. From this page, your customers will be able to download and run Triage Client. Triage Client will collect all the information needed to diagnose problems with your application, and e-mail it back to you.

Profile an application

A profile specifies what diagnostic information to collect for an application. **Auto Profile** lets you automatically generate a profile from:

- A Windows Installer package (.msi) file.
- An InstallShield, Wise Installer, or Visual Basic project.
- An installed application.

For this quickstart, we will use an installed application.

To auto-profile an application:

- 1 In the **Action** menu, click **New** and then click **Profile**.

- 2 In the profile tree, click **Auto Profile**.

- 3 Click **Application**.

Triage displays a list of applications found on your computer.

- 4 Click an application and click **Ok**.

Triage starts the application, audits your computer, and generates the profile.

- 5 Save the profile.

Save Profile



After you finish auto-profiling your application, you'll see that the profile now contains lists of items such as files, ActiveX controls, registry entries, shortcuts, and files to copy. The actual contents of the generated profile depend on the application.

Generate Repair Rules and a Self-Repair package


This step is optional.

Repair rules specify how to identify problems with an application. For example, a self-repair rule might look like this:

```
if ( audit status = found and size = 33845 )
then
    Do Nothing
else
    Fix it
```

A self-repair package is an archive (a .ZIP file) of the files needed to fix problems. For example, a self-repair package contains the required versions of application DLLs.

To generate repair rules for all items

- 1 In the console tree, click a profile.
- 2 In the Details view toolbar, click .
- 3 Click **Yes** to perform an audit of the local computer. Triage Console uses the information gathered during the audit to build the repair rules.
- 4 Type a URL for the self-repair package. This is location from which Triage Client downloads the package. For example, "http://www.server.com/app-repairs.zip". You need to include a file name. Triage Console will use this file name for the package, which it saves in the same location as the profile.

Triage creates self-repair rules for all items (files, ActiveX controls, registry entries, ...) in the profile, and a self-repair package containing all the files and ActiveX controls.


To view or edit a generated repair rule:

- 1 In the profile tree, click **Files, Registry, ActiveX Controls, Environment Variables, or Shortcuts**.
- 2 Click an item (a file, ActiveX control, shortcut, environment variable, or registry entry).
- 3 Click **Self Repair** and click **Build Condition**.

Create an eSupport package

An e-Support package contains everything you need to provide automated technical support over the Web.

To generate an eSupport package:

- 1 Open an existing profile or create a new profile.
- 2 In the Profile toolbar, click **Package for WEB distribution** .
- 3 Select the folder where you want Triage to put the eSupport package and the HTML pages and other eSupport files.

Triage Console generates the eSupport package and copies all the eSupport files (such as the HTML pages) to a subfolder of the specified folder. The subfolder is given the same name as the profile.

You can put the generated HTML pages on your Web site so your customers can download the .ZIP file and run Triage Client. Note that your customers don't need to know anything about Triage Client or ZIP files. They just have to click a button.

What Next?

This quick start shows you how to do 80% of the work. You'll probably want to do some fine-tuning, such as the following:

Digitally sign your e-Support packages If you want to use Netscape, you need to install the digital certificate before you generate the e-Support package.

Customize the HTML page templates For example, to insert your company and application names.

Set the eSupport audit preferences For example, the **Web E-mail Address** preference specifies the default e-mail address used by Triage Client when it sends audit reports.

Chapter 2: Building Profiles

A profile specifies what diagnostic information you want to collect, either for a software application or a hardware peripheral. Diagnostics information can include:

- Files, registry entries, ActiveX controls (.ocx), self-registered files (.ocx or .dll), shortcuts, and environment variables.
- System, operating system, and hardware information.
- Copies of text and binary files. For example, you can get copies of text files such as .INI, .SYS, and .BAT files.

You can automatically generate a profile from:

- A Windows Installer package (.msi) file.
- An InstallShield, Wise Installer, or Visual Basic project.
- An existing installation.

You can also use Triage Console to manually edit the details of a profile, or to manually build a profile.

Profiling and the Windows Operating System

If you need to diagnose problems on multiple versions of Windows, build one profile that covers all the Windows versions you need to support. It is easier to maintain and distribute a single version of a profile.

Building Profiles

Triage provides several methods for building profiles:

Import a Windows Installer package (*.MSI) file This is the recommended way to build a profile.

Import the project file for an install package If you use InstallShield (5.x, 6.x), InstallShield Express, or Wise Installer to develop install programs for your applications, you can import the project files. You can also import Visual Basic projects.

Generate the profile from an application installation when an application doesn't use Windows Installer and you don't have the source files for the setup, you can use an existing installation of the application.

Build the profile manually This method is ideal for building profiles for collecting configuration information for performing system change analysis. For example, to troubleshoot problems with hardware components such as printers and video cards that have associated software.

For applications, building profiles manually requires considerable, detailed knowledge of the application.

Auto-profiling Applications

Specifying What to Include The **Files, Registry Entries, Self-registered files** (for example, OCXs), and **Shortcuts** check boxes control what items are included in the generated profile.

Filtering Out Files and Registry Keys As a general guideline, a profile should not exceed 1 megabyte in size. To control the size of a profile, use:

- File extensions to ignore to specify which files you do not want to include in the profile.
- Registry keys to ignore to specify which registry keys (for example, HKEY_CLASSES_ROOT) you do not want to include in the profile.

Filters are specified as a comma-separated list. You can include one or more spaces between commas to make the list more readable.

Copying Files When you generate a profile automatically, you can generate a list of files (ASCII or binary) to retrieve from the remote PC. For example, you can retrieve .INI and other configuration files from a user's PC.

The **File extensions to process as Copy Files** box is a comma-separated list of file extensions. When the profile is generated, all files with these extensions are added to the list of files to copy.

If the install path is found in the registry, Auto Profile creates a variable for the application install directory.

Importing Windows Installer (MSI) Packages

Microsoft Windows Installer is a component of the Windows operating system that manages the installation and removal of applications. A package (.MSI) file stores information regarding the application setup and installations and is distributed to end users.

Generating a profile from an MSI file is more reliable than reverse engineering an existing installation of the application. Whenever an application uses Windows Installer, you should use its MSI file to create its profile.

To import an MSI package:

- 1 In the console tree, right-click **Profiles**, click **New**, and then click **Profile**.

This creates a public profile (a profile that is available to all users running a Triage Console). To create a private profile, expand **Profiles**, right-click **Private**, click **New**, and then click **Profile**.

- 2 In the profile tree, click **Auto Profile**.
- 3 Click **MSI**. Triage displays a list of the MSI packages found on your system (in the Installer subfolder of your Windows System folder, for example, C:\WinNT\Installer).
- 4 If you don't see the package you want, click **Browse** to locate it.
- 5 Click a package and click **OK**.
- 6 Select the features you want to import into the profile.

For example, the top-level features of Microsoft Office are Word, Excel, Power Point, and so on. This allows you to create separate profiles for each Office program instead of one large monolithic profile for all of Microsoft Office.

For each top-level feature, you may also want to create profiles with and without optional features that some users may not install. For example, you may want a profile for an installation of Microsoft Word without the spell checker, so that all the profile items related to the spell checker won't be and identified as problems.

MSI files typically have unfriendly names such as 4499fdf.MSI. To find the MSI file you want, point to the file until the tooltip appears, or add the Title column to the Details view of the dialog (right click a column header, click More, and select the Title check box).

Avoid building large profiles.

MSI Files and Self-Registered DLLs

When you import an MSI file, the self-registered DLLs are not listed in the ActiveX Controls section of the profile. Instead, the Registry section includes all the registry entries required by the DLLs.

Importing Install Packages

If you have the source for an install package, you can use it to build a profile. Triage can automatically import items from the following common install packages:

Install Package	What you can import
InstallShield 5.x, 6.x InstallShield 5.x, 6.x Log File Wise Installer	Files Self-registered files Registry keys Shortcuts
InstallShield Express	Files Self-registered files Registry keys
Visual Basic Project (vbp)	Files Self-registered files

To import an install package:

- 1 In the console tree, right-click **Profiles**, click **New**, and then click **Profile**.

This creates a public profile (a profile that is available to all users running a Triage Console).

To create a private profile, expand **Profiles**, right-click **Private**, click **New**, and then click **Profile**.

- 2 In the profile tree, click **Auto Profile**.
- 3 Click **Package**.
- 4 In the **Files of type** box, select the type of install package you want to import.
- 5 Click an install package and click **OK**.

To import an InstallShield 5.x log file:

- 1 Load the log (ISU) file in the InstallShield 5.x log file viewer.
- 2 Save it as a text file. Triage Console can read only the text version of the log file.

Installed Applications

Triage can generate a profile from an existing installation of an application. After you select an installed application, Triage scans your system for information about the application (such as files, registry entries, and shortcuts), starts the application to determine what ActiveX controls it uses, and then generates the profile.

To auto-profile an installed application:

- 1 In the console tree, right-click **Profiles**, click **New**, and then click **Profile**.

This creates a public profile (a profile that is available to all users running a Triage Console). To create a private profile, expand **Profiles**, right-click **Private**, click **New**, and then click **Profile**.

- 2 In the profile tree, click **Auto Profile**.
- 3 Click **Application**. Triage displays a list of applications found on the local PC.

If you do not see the application you want to profile in the **Installed Applications** dialog, click **Browse** and locate the application executable on your computer.

- 4 Click an application and click **OK**.

ActiveX Controls Triage can determine only the ActiveX controls loaded at startup. ActiveX controls loaded on demand by the application are

not included in the generated profile. If you are familiar with the application, you can manually add the missing ActiveX controls.

Too Many Files? If the generated list of files is too large, add some file extensions to the **File extensions to ignore** filter and generate a new profile.

Files with No Path If a file is listed with no path, it was probably found somewhere on your hard disk outside of the application installation directory and the standard Windows directories (for example, c:\temp). Generally, you can remove such files from the profile.

After You Auto Profile

- Because not all applications follow standard rules for installations, profiles for installed applications may not be complete. Visually inspect the profile and verify that the files, registry entries, shortcuts, and so on make sense.
- If the profile includes keys or values under HKEY_CURRENT_USER, the user must be logged on when Triage Client audits the user's computer. Otherwise, if no one is logged on, the current user will be the default user and the audited values will not reflect the user's environment.
- Make sure all paths to files and shortcuts use variables.
- If the application depends on environment variables, you must add them by hand.
- Check the Product attributes.

Manually Building a Profile

To manually build a profile, you have to decide what information you want to collect. For example:

- Do you want to collect information on files? Which files? DLLs, ActiveX controls, shortcuts, or other types of files? Do you want to retrieve copies of files?
- Do you want to check the registry for specific keys and values?
- What kind of system configuration information do you want to collect? Installed applications? Running services? Loaded modules? Memory usage? Hardware components?

Adding Items

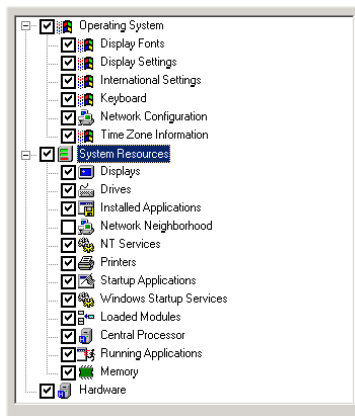
Triage Console includes tools for building lists of items you want to audit, such as files, ActiveX controls, registry keys and values, shortcuts, and environment variables. To simplify the process, you can use regular expressions to select groups of files based on their names (for example, all the DLLs in a folder). You can also define variables to represent computer-specific values such as paths.

See Chapter 4, “Collecting Diagnostics and Configuration Information” on page 17 for more information on adding items to a profile.

Collecting System Resource Information

Setting up a profile to collect system resource information is straightforward. Just check off the items you want to collect.

System Resources



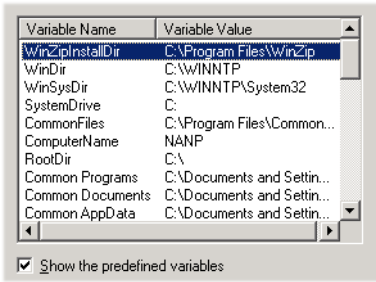
System resource information can include:

- Operating system information.
- System configuration information such as the amount of free disk space, what DLLs are loaded into memory, and what applications are running.
- Hardware component and configuration information.

Defining Variables

You use variables to represent paths that can vary from computer to computer, such as the location of the Windows system folder or the installation folder of an application.

Variable Definitions

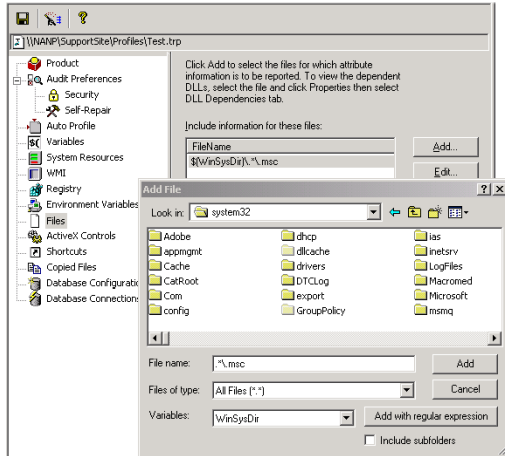


If you want to collect information on files and shortcuts, or retrieve copies of file, you can use variables to locate the files on each computer.

For example, you can use variables to represent the location of the Windows system folder, the installation folder of an application, or the location of the shortcuts on the **Start** menu.

For application files, you can define a variable that extracts the application install path from the registry, or use a predefined variable such as \$(Common Files), which stores the location of the Program Files\Common Files folder.

Adding Files



Chapter 3: Performing Change Analysis

Change analysis is a basic technique for troubleshooting system and application problems. It's the process of tracking down configuration changes on a computer.

With Triage, you can build profiles to collect application and system configuration information, audit computers, and then analyze the collected diagnostic data. Triage automatically compares application or system settings against a baseline or at different points in time. This allows you to quickly identify the changes that caused the problem.

To perform change analysis, you compare an audit report generated on a user's machine with a reference audit report. A reference report is a baseline audit report generated on a machine where your application runs without any problems.

Creating a Reference Audit Report

To create a reference audit report that you can use as a baseline, run Triage Client on a computer where a working version of the application is installed. Then add the reference audit report to the **Public** audit reports in Triage Console.

Adding Audit Reports to Triage Console

To view audit reports or perform change analysis, you must make the audit reports available in Triage Console.

To add an audit report to the console tree:

- 1 In the console tree, right click **Audits** and click **Open File**. Locate the audit report (.zip or .tra) you want to open and double-click it.
- 2 Click **Yes** to add the audit report to the public audit reports, or click **No** to add the audit report as a private report.

Public audit reports are stored in the SupportSite (in \\server\SupportSite\AuditReports) and are available to all users running a copy of Triage Console.

You can add public audit reports by copying .tra or .zip files to the SupportSite\AuditReports folder, or by saving the audit report from Triage Client directly to the SupportSite\AuditReports folder.

In Triage Console, you may need to refresh the console tree (right-click Public and then click Refresh).

Private audit reports are stored outside of the SupportSite\AuditReports folder, for example on your local hard disk.

Viewing Audit Reports

To view an audit report:

- 1 In the console tree, expand **Audits**, expand **Private** or **Public**, and click an audit report.



- 2 In the Details view, expand the sections of the audit report you want to view.

If a section name is highlighted in a different color, that means an item is missing in the audit report (for example, a file was not found).

*In an audit report, the **Variables** section contains the values of the variables on the audited computer.*

Opening and Editing Copied Files

By default, ASCII and binary files are always attached to the audit report, and opened or edited with their associated applications. However, ASCII files can be included in the body of the audit report, and viewed directly in Triage Console (if the **Attach Copied Files** preference is set to **False**). Including copied files in audit reports also allows you to compare the contents and highlight differences.

To view attached files:

- 1 In the Details view, expand **Copied Files**.
- 2 Under **Copied Files**, right-click the file you want to view.
- 3 Click **Open**, **Open With**, or **Edit**.

The command you choose depends on the type of file and what actions are associated with that file type. For example, on some systems, **Open** will execute a javascript (.JS) file, while **Edit** will simply load the file into a text editor.

If you are not sure, click **Open With** and click the program you want to use to open the file.

To view included files:

- 1 In the Audit view, expand **Copied Files**.
- 2 Under **Copied Files**, expand the file you want to view.
- 3 Expand **Contents**.

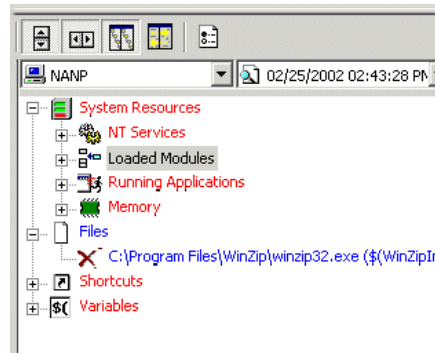
To copy content from included files:

- 1 Expand **Contents**.
- 2 Right-click the line you want to copy and click **Properties**.
- 3 Highlight the text you want to copy.
- 4 Right-click the highlighted text and click **Copy**.

Comparing Audit Reports

When you compare two audit reports, Triage automatically highlights any differences between the two reports. This allows you to review configuration changes and quickly spot bad or missing files, wrong file versions, missing registry entries, invalid OS settings, and more.

Changes Visually Highlighted







You can compare a computer's configuration:




- Against a baseline.
- At two different points in time.


*The **reference report** is in the left pane, and the **audit report** is in the right pane.*

To compare audit reports:

- 1 In the console tree, click the audit report you want to use as a baseline for the comparison.
- 2 In the Details view toolbar, click **Compare Audit Reports** .
- 3 In the right pane, click an audit report in the list of available audits.
- 4 Review the differences:
 - Click  to display only the differences.
 - Click  or  to display the **next or previous** difference.

By default, the display of the two audit reports is synchronized, so that both reports scroll up and down together, and expand and collapse together. This makes it easier to perform a side-by-side comparison of the reports. Turn this feature off if you want to view each report independently..


To turn off	Click
Synchronized vertical scrolling	
Synchronized horizontal scrolling	
Synchronized expanding and collapsing of report sections	

Click **Synchronize Item**  to display the same item in both reports when display synchronization is turned off.

Filtering Audit Reports

Filtering allows you to filter out irrelevant differences when comparing audit reports. Use filters to reduce the number of differences displayed when you view differences only.

To filter out differences:

- 1 In the Details view toolbar, click **Options** .
- 2 In the **Filters** tab, clear the check boxes for the audit items you want to filter out.
 - Select when to apply the filter: When viewing differences only or all items, click **Always**.
 - When viewing differences only, but not when viewing all items, click when viewing **Differences Only**.

To disable filtering, click **Never** on the **Filters** tab.


Triage saves the filter settings, so each time you compare two audit reports the same items are filtered out.

Filtered items are never highlighted when they are different. For example, if you choose to always apply a filter, the filtered items are never highlighted as different, even if they are.

Filters are ignored if you load a single report.

Customizing the Difference Highlighting

To customize difference highlighting:


- 1 In the Details view toolbar, click **Options**  and then click the **General** tab.
- 2 Change the colors.

To change the color of	Do this
Items that are different in each report.	In the Color of different items list, click a color.
Items missing in the audit report displayed in the left pane	In the Color of items missing in reference report list, click a color.
Items missing in the audit report displayed in the right pane	In the Color of items missing in audit report list, click a color.

Hiding Files from Non-Active Operating Systems

When more than one operating system is installed on a computer, an audit report contains information for each operating system. You can filter out the non-active operating system when viewing the audit report.


To filter out files from the non-active OS:

- 1 In the Details view toolbar, click **Options**  then click the **General** tab.
- 2 Click **Ignore files in the non-active operating system**.

Printing Audit and Diagnostic Reports


Triage can print audit reports and diagnostic reports. A **diagnostic report** summarizes the differences between two audit reports. You can also save diagnostic reports (in a .TRD file).

To print an audit report:


- 1 View an audit report.
- 2 In the Details toolbar, click  and then click **Print Reference**.

If you are comparing audit reports, **Print Reference** prints the audit in the left pane, and **Print Audit** prints the audit in the right pane.

To print a diagnostic report:

- 1 Compare two audit reports.
- 2 In the Details toolbar, click  and then click **Print Diagnostic**.

To save a diagnostic report:

- 1 Compare two audit reports.
- 2 In the Details toolbar, click  and then click **Print Diagnostic**.

Chapter 4: Collecting Diagnostics and Configuration Information

In addition to collecting information on files, registry entries, ActiveX controls, self-registered files, shortcuts, and environment variables, a profile can also collect:

- System, operating system, and hardware information.
- Copies of text and binary files. For example, you can get copies of text files such as .INI, .SYS, and .BAT files.
- Database configuration and connection information.
- Advanced diagnostics from Microsoft Windows systems through Windows Management Instrumentation (WMI).
- Diagnostic information about Microsoft Internet Information Server (IIS).

Defining Variables

Triage uses variables to specify the paths to files and shortcuts. A variable can represent a file path that can vary from machine to machine. For example, the location of the Windows folder can vary from machine to machine, and different users can install an application in different directories.

If an application stores paths in the registry, in an INI file, or relies on environment variables, Triage can use variables to look for files and shortcuts only in those locations. Otherwise, Triage searches the entire computer. Similarly, if you know that a file should be in the Windows folder, you can use a variable to search only the Windows folder.

Variables can be combined together to form a single expression. Variables can also be combined with regular expressions.

Triage provides the following variable types:

- **Registry variables** that are expanded based on a value stored in the registry.
- **INI variables** that are expanded based on a value stored in an INI file.
- **Predefined variables** that are automatically expanded by Triage.
- **User-defined variables**, which are expanded based on values entered by a Triage Client user.
- **Environment variables** such as Path and TEMP.

Using Variables

To reference a variable, you type an expression of the form \$(Variable Name), where Variable Name is the name you gave to the variable when you defined it.

To reference an environment variable, enclose it in “%(“ and “)”. For example, “%(TEMP)”.

You can use variables with the following items:

- File names of files, shortcuts, ActiveX controls, and files to copy (to specify computer-specific paths)
- Definitions of variables

You can use INI, Registry, Pre-defined, and Environment variables in the definitions of INI and Registry variables.

- Values of the **Pre-audit Application** and **Post-audit Application** audit preferences.
- Self-repair rules for files, ActiveX controls, shortcuts.
- Repair rule conditions.
- Database Information such as the: database connection names, SQL statements, SQL server attributes, and SQL connection strings.

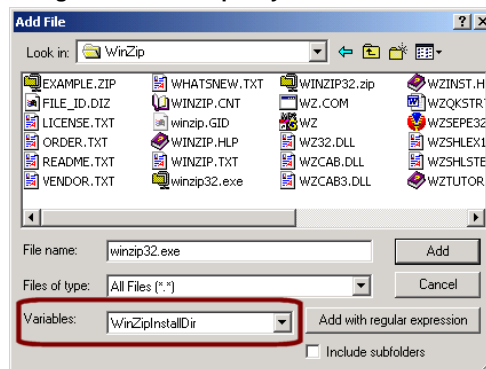
If you use a user-defined variable, you can prompt the user for input just before the audit.

Registry Variables


A registry variable represents a value stored under a registry key (either the default value or a named value). You use these variables to specify file paths when you add files, ActiveX controls, files to copy, or shortcuts to a profile. When you use a variable to specify the path to a file or shortcut, Triage looks only in that location for the file. Otherwise, Triage searches all drives for the file.

For example, suppose an application stores its installation directory in the registry as the default value of a key named InstallPath. If you want Triage to look for files in this installation directory, you can define a variable that extracts the default value of the InstallPath key. Then you can use this variable to specify the location of the file.

Using a Variable to Specify the Location of a File



To define a registry variable:

- 1 In the profile tree, click **Variables**.
- 2 Click **Add**.
- 3 Click **Registry** to define a registry variable.
- 4 In the **Registry Key** row, click  to open the Registry dialog, and select a registry value.

If you select a registry key, the variable is given the default value of the key (if the default value is set).

- 5 In the **Variable Name** row, click in the **Value** column and enter a name for the variable.

If Interactive Mode is True, Triage Client prompts the user to enter a value if the registry key is not found. Prompt Message is the message displayed by Triage Client when it prompts for input.

INI Variables

An INI variable represents a value stored in an INI file. You can use these variables to specify file paths when you add files, files to copy, or

shortcuts. For example, suppose an application stores its installation directory in an INI file as follows:

```
[Paths]
```

```
InstallPath=C:\Program Files\Company\App
```

You can then define an INI variable that extracts the value of the InstallPath entry in the PATHS section of the INI file. This INI variable can then be used to specify the location of a file.

When you use a variable to specify the path to a file, Triage looks only in that location for the file. Otherwise, Triage searches all drives for the file and audits every instance it finds.

To define an INI variable:

- 1 In the profile tree, click **Variables**.
- 2 Click **Add**.
- 3 Click **INI** to define an INI variable.
- 4 Enter the name of the INI file, the name of the INI section, and the name of INI entry.
- 5 In the **Variable Name** row, click in the **Value** column and enter a name for the variable.

If Interactive Mode is True, Triage Client prompts the user to enter a value if the registry key is not found. Prompt Message is the message displayed by Triage Client when it prompts for input.

User-Defined Variables

A user-defined variable represents a named value that is requested from the user when Triage Client audits a system

A user-defined variable can also be used as a constant. For example, if you want to use the same value (for example, a string) in a number of places, you can define a variable to hold this value.

To define a user-defined variable:

- 1 In the profile tree, click **Variables**.
- 2 Click **Add**.
- 3 Click **User Defined**.
- 4 In the **Variable Name** row, click in the **Value** column and type a name for the variable.
- 5 Type a **Prompt Message** to display when prompting the user for input.
- 6 Set the **Variable Data Type**. This determines how Triage Client prompts users to enter the variable value.

File	File selection dialog.
Folder	Folder selection dialog.
Memo	Multi-line text box.
Registry	Registry Viewer dialog.
Text	Single-line text box.
- 7 In the **Variable Value** row, enter a value for the variable. This allows you to use the variable while building the profile. For example, if the variable represents a path, you can use it to add files.
- 8 Set **InteractiveMode** to **True**.
- 9 In the **Prompt Message** row, type the text you want to display on the input dialog.

To define a constant:

- 1 In the profile tree, click **Variables**.
- 2 Click **Add**.
- 3 Click **User Defined**.
- 4 In the **Variable Name** row, click in the **Value** column and type a name for the variable.
- 5 In the **Default Value** row, click in the **Value** column and type a value. This is the value of the constant.

Other Variable Attributes

The **Variable Value** attribute is set when you click **OK** or **Apply**. This value is used while building the profile (for example, to find the files you add to the profile). During an audit, the variable value is determined by the settings of the user's computer.

The **Default Value** attribute is used when the value cannot be extracted from the INI file. For example, during an audit, the **Default Value** attribute is assigned the value of the variable. Then when the application needs to be repaired, the value will be available even if it cannot be found in the registry.

The **Extract As** and **Variable Data Type** attributes are used to extract folder paths from file names. See "Extracting Folders from File Names" on page 20.

The **Interactive Mode** attribute determines whether Triage Client prompts the user for input when the variable has no value.

Extracting Folders from File Names

Sometimes, an application does not store its installation path in the registry, but it does store the full path names of some files in its installation folder. You can define a variable that gets the file name from the registry, and then extracts only the path part.

For example, if a registry value is **C:\Program Files\MyApp\myapp.exe**, you can define a variable that extracts just the **C:\Program Files\MyApp** part.

To extract the folder from a file name:

- 1 Create a new registry variable.
- 2 Set the **Extract As** attribute to **Folder**. This specifies how to extract the variable value when replacing a variable reference in the profile.
- 3 Set the **Variable Data Type** attribute to **File**. **Variable Data Type** specifies what kind of value is stored in the registry key.

For example, if you auto profile the WinZip application, the following variable is defined:

Variable Name	WinZipInstallDir
Extract As	Folder
Registry Key	HKCU\software\nico mak computing\winzip\programs\zip2exe
Variable Value	G:\Program Files\WinZip\WZSEPE32.EXE
Variable Data Type	File

Prompting Users for Input

Triage Client can prompt users to enter values for registry, INI, and user-defined variables.

For example, if the installation path for an application is not stored in the registry or in an INI file, you can use a user-defined variable to represent the installation path when you set up the profile. Before auditing the user's system, Triage Client pops up a Select Folder dialog so the user can select the actual installation path.

Or if a registry key or INI entry is not found on a user's machine, you can prompt the user to enter a value for the variable.

To prompt users to enter variable values

- 1 For registry and INI variables, set **Interactive Mode to True**.

If the registry key or INI entry is not found, Triage Client prompts the user to enter the variable value.

- 2 Type a **Prompt Message** to display when prompting the user for input.
- 3 Set the **Variable Data Type**. This determines how Triage Client prompts users to enter the variable value.

File	File selection dialog.
Folder	Folder selection dialog.
Memo	Multi-line text box.
Registry	Registry Viewer dialog.
Text	Single-line text box.

- 4 Set the **Variable Value**.

The **Variable Value** attribute is used only for building the profile. For example, if the variable did not have a value, then when you used it to add a file Triage Client would not be able to find the file.

Predefined Variables

Predefined variables are variables whose values are supplied by Triage when you audit a computer. Most of the predefined variables provide computer-specific values, such as the location of the Windows folder and the name of the computer.

Predefined System Variables

WinDir Windows folder (for example, "c:\WinNT").

WinSysDir Windows system folder (for example, "c:\WinNT\system32").

SystemDrive Drive where the operating system is installed (for example, "c:\").

ClientDir Folder where Triage Client is located.

CommonFiles Windows common files folder (for example, "c:\Program Files\Common Files").

ComputerName Name of the computer (for example, "KIMA").

RootDir Boot drive (for example, "c:\").

Predefined User-profile Variables

Common Desktop Location of the shared Desktop folder. For example, %SystemRoot%\Profiles\All Users\Desktop.

Common Documents Location of the shared Documents folder. For example: C:\Documents and Settings\All Users.WINNT\Documents.

Common Administrative Tools Location of the shared Application Data folder. For example, C:\Documents and Settings\All Users\Administrative Tools\.

Common AppData Location of the shared Application Data folder. For example, C:\Documents and Settings\All Users\Application Data\.

Common Programs Location of the shared Programs folder. For example, %SystemRoot%\Profiles\All Users\Start Menu\Programs.

Common Start Menu Location of the shared Start Menu folder. For example, %SystemRoot%\Profiles\All Users\Start Menu.

Common Startup Location of the shared Startup folder. For example, %SystemRoot%\Profiles\All Users\Start Menu\Programs\Startup.

Common Templates Location of the shared Templates folder. For example, C:\Documents and Settings\All Users\Templates\.

Personal Location of the current user's My Documents folder. For example, C:\Documents and Settings\stephen\My Documents\.

AppData Location of the current user's Application Data folder. For example, C:\Documents and Settings\stephen\Application Data\.

Cookies Location of the current user's Cookies folder. For example, C:\Documents and Settings\stephen\Cookies\.

Desktop Location of the current user's Desktop folder. For example, C:\Documents and Settings\stephen\Desktop\.

Favorites Location of the current user's Favorites folder. For example, C:\Documents and Settings\kima\Favorites\.

NetHood Location of the current user's NetHood folder. For example, C:\Documents and Settings\kima\NetHood\.

My Pictures Location of the current user's My Pictures folder. For example, C:\Documents and Settings\kima\My Documents\My Pictures\.

PrintHood Location of the current user's PrintHood folder. For example, C:\Documents and Settings\kima\PrintHood\.

Recent Location of the current user's Recent folder. For example, C:\Documents and Settings\kima\Recent\.

SendTo Location of the current user's SendTo folder. For example, C:\Documents and Settings\kima\SendTo\.

Start Menu Location of the current user's Start Menu folder. For example, %SystemRoot%\Profiles\%UserName%\Start Menu.

SendTo Location of the current user's SendTo folder. For example, C:\Documents and Settings\kima\SendTo\.

Templates Location of the current user's Templates folder. For example, C:\Documents and Settings\kima\Templates\.

Startup Location of the current user's Startup folder. For example:

%SystemRoot%\Profiles\%UserName%\Start Menu\Programs\Startup.

Local Settings Location of the current user's Local Settings folder. For example, C:\Documents and Settings\kima\Local Settings\.

Local AppData Location of the current user's local Application Data folder. For example, C:\Documents and Settings\kima\Local Settings\Application Data\.

Cache Location of the current user's Temporary Internet files folder.

History Location of the current user's History folder.

Fonts Location of the system fonts folder. For example, C:\WinNT\Fonts.

Administrative Tools Location of the current user's Application Data folder. For example, C:\Documents and Settings\kima\Administrative Tools\.

Predefined Generated Variables

UniqueFilename. A unique filename generated from a combination of the user's name, the product name, and a time stamp.

Predefined Audit Report Variables

Audit Report Path Location of the audit report on the user's machine.

Audit Report Filename Name given to the audit report by the user when the report is saved, posted to the web, or sent via e-mail.

You use the \$(Audit Report Filename) variable to set the value of the **InvokeServlet** web publishing attribute.

Environment Variables

You can control where Triage locates files by prefixing a filename with an environment variable. For example, to locate a file in the **TEMP** directory, you can specify **%(TEMP)\myfile.txt**.

During an audit, Triage Client first tries to get the value from a user environment variable. If it cannot find a user environment variable with the specified

name, it tries to get the value from a system environment variable. During self-repair, Triage Client only looks at system environment variables

Using Regular Expressions

Use regular expressions to select groups of files based on their names. For example, to select all MFC DLLs in the Windows system directory, you would use the regular expression `"^mfc.*\dll"`.

Triage audits any file whose name contains a substring that matches the regular expression. So, for example, the regular expression `"mfc"` matches any file containing the string `"mfc"`—not just the DLLs, but also files like `"mfcuix.hlp"` and `"MFC Tracer"` (a shortcut).

. The period (.) matches any character. For example, `"ie."` matches both `"ie5"` and `"ie6"`. To match an ordinary period, you use the backslash. For example, `".\ini"` matches `".ini"`.

***** The asterisk (*) matches zero or more occurrences of the preceding character. For example, `".*" matches any string of characters, and ".*\dll" matches all DLLs.`

^ The caret (^) matches the beginning of a string. For example, `"^reg"` matches any string that begins with `"reg"`.

\$ The dollar sign (\$) matches the end of a string. For example, `"ini$" matches any string that ends with "ini". And while ".\ini" matches both "runlog.ini" and "foo.init", ".\ini$" matches only files with a ".ini" extension.`

[] Matches a range of characters. For example, `"[A-Za-z0-9]" matches any alphanumeric character. "[0-9]" matches zero or more digits. If`

the first character is the caret (^), the expression matches any character not in the range. For example [^AB^] matches any character except A, B and the caret itself.

\ Used to escape special characters. For example, "\." matches a period (.) and "\\$" matches a dollar sign (\$).

Examples

To look for all files that have a .DLL extension:

- 1 In the **File Name** box, type the regular expression "*.DLL".
- 2 Click **Add with regular expression**.

To look for all files in a specific folder:

- 1 In the **File Name** box, type the regular expression ".*\.*".
- 2 Click **Add with regular expression**.

To look for all files that have a .DLL extension in the Windows system directory:

- 1 In the Variables list, click the **WinSysDir** variable.
- 2 In the **File Name** box, type the regular expression "*.DLL".
- 3 Click **Add with regular expression**.

To look for all files that have a .DLL extension in the Windows system directory and its subfolders:

- 1 In the Variables list, click the **WinSysDir** variable.
- 2 Click the **Include subfolders** check box
- 3 In the **File Name** box, type the regular expression "*.DLL".

- 4 Click **Add with regular expression**.

To look for all files that have a .DLL extension in a subfolder of the Windows system directory:

- 1 In the Variables list, click the **WinSysDir** variable.
- 2 In the **File Name** box, type the regular expression "aSubFolderName\.*\DLL".
- 3 Click **Add with regular expression**.

System Resources

Triage can collect a wide variety of information about the configuration of a computer:

- System resource information, including displays, drives, installed applications, NT services, printers, startup applications, loaded modules, central processor, running applications, memory, and RAM.
- Operating system information, such as international settings, keyboard, time zone information, and Windows system information.
- Hardware information about components such as CD-ROM drives, disks, displays, hard drive controllers, monitors, ports, and system boards.

To collect system resource information:

- 1 In the profile tree, click **System Resources**.
- 2 Select the check boxes for the information you want to collect. Clear the check boxes for information you don't want to collect.

*To select just one or two check boxes under **Operating System** or **System Resources**, clear the top-level check box. This clears all check boxes so you can then select the check boxes you want.*

By default, the *Network Neighborhood* check box (under *System Resources*) is cleared. Do not select this check box for large networks because auditing can take a substantial time.

The system resource information collected by Triage depends on the version of Windows installed. For example, *Display Fonts* information is collected on Windows 95 and 98, but not on Windows NT or 2000. If Triage does not collect the system resource information you need, use *Windows Management Interface (WMI)* to collect the required information. See “Auditing with *Windows Management Instrumentation*” on page 28.

Auditing Files

A profile includes a list of application files that you want to audit. To include files in a profile, you can:

- Select files from the folders on your computer or on any other computer in the network neighborhood.
- Add all DLLs that one of your application DLLs depends on.

For an EXE file, Triage automatically collects information about the DLLs that the EXE loads (so you don't have to add the DLLs yourself in Triage Console).

But if you want to collect information for all instances of a DLL on a system, you must add the DLL to the profile.

When you audit a file, Triage collects information for all instances of the file found on the computer. Use variables in the file name to collect information for only one specific instance of the file.

To add files:

- 1 In the profile tree, click **Files**.
- 2 Click **Add**.
- 3 Locate the folder containing the files you want to add.
- 4 Add the files you want to audit:

To add specific files, select the files.

To add all files whose names match a regular expression, type the regular expression in the **File Name** box.

- 5 If you have defined a variable to represent the location of the files, then in the **Variables** list, click the variable that represents the location of the files.
- 6 If you selected the files, click **Add**. If you typed a regular expression in the **File Name** box, click **Add with regular expression**.

To use a variable to represent the location of the files:

In the **Variables** list, click the variable.

Note that if you use a variable and regular expressions, you do not have to locate the actual folder containing the files.

To search subfolders for the files:

Click the **Include Subfolders** check box.

To include files in a profile even if they do not exist on your computer:

Type the file names in the **File Name** box.

To search network drives and CDROMs:

By default, Triage searches for files on the local hard drives of a user's machine. If you want Triage to also search network or CD-ROM drives by default, set **Include Network Drives** and **Include CDROMS** to **True** in the **Audit Preferences**.

To add DLL dependencies:

- 1 Add a DLL to the profile, click it, and then click **Properties**.
- 2 Click the **DLL Dependencies** tab to browse the hierarchy of DLLs that your application DLL depends on.
- 3 Click **Add All** to add all the required DLLs to the list at the bottom of the dialog, or click **Add Selected Item** to add just the selected DLL.
- 4 Click **OK** to add the DLLs to the profile.

Collecting File Version Information

The **File Version Information** audit preference determines how much file version information is collected during an audit. Setting this attribute to **Minimal** or **Normal** reduces the amount of memory and time required to audit files. It also reduces the size of the audit reports, so they load and compare faster.

Minimal extracts FileVersionProp, FileDescriptionProp, and LegalCopyrightProp.

Normal extracts the Minimal information plus: CompanyName, InternalName, OriginalFileName, Productname, and ProductVersion.

Full extracts Normal and Minimal information plus: Comments, FileVersion (not the same as the one above), ProductVersion (not the same as the

one above), TradeMarks, PrivateBuild, Special-Build, fileFlagsMask, FileFlags, Os, Type, SubType, Translations, and TranslationsCharset.

Auditing ActiveX Controls

A profile can include a list of ActiveX controls (.OCX) and self-registered files (.OCX or .DLL) to audit. For example, you can set up a profile to check that a DLL is registered correctly.

For each ActiveX control listed in the profile, an audit report includes the CLSID and TypeLib information found in the registry, as well as general and file version information.

To add ActiveX controls to a profile:

- 1 In the profile tree, click **ActiveX Controls** and then click **Add**.
- 2 In the **Add ActiveX Controls** dialog, select the files you want to add and click **Add**. You can also type the name of a file in the **File Name** box.

Auditing Registry Keys and Entries

A profile can include a list of registry keys and values to collect during an audit.

Adding Keys and Values

If you add a registry key, Triage adds all values and subkeys under that key, and selects the key. If you add a registry value, Triage adds just the value and selects it.

Selecting Keys and Values

During an audit, Triage gets the selected keys and values. To select a key or value, click the check box for the key or value.

For each selected key, Triage gets all values entered in the registry for the key. If the **Recursive Registry Scan** audit preference is **True**, Triage gets all subkeys and values under that key.

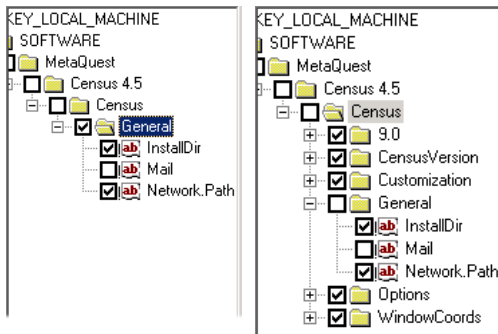
Only selected keys and values can be repaired.

Synchronizing

Synchronizing allows you to add missing subkeys and values. For example, after manually adding a single subkey, you may decide you want all the subkeys at the same level. To do this, click the parent key and then click **Synchronize**.

After you synchronize, you must select the keys and values you want to audit.

Before and After Synchronizing a Key



Restricting Keys

To prevent users from selecting keys such as **HKEY_LOCAL_MACHINE\Software** and all their subkeys and values, you can build a list of restricted keys. Restricted keys cannot be added or selected.

The list of restricted keys is stored in the file **ProfViewer.ini**, which you can find in the Triage installation folder.

Auditing Shortcuts

A profile can include a list of shortcuts (.LNK files) to check. For example, you can set up a profile to check that a shortcut exists and that it points to the correct target.

For each shortcut listed in the profile, an audit report includes shortcut properties such as the shortcut's target, arguments, and working directory.

To add shortcuts to a profile:

- 1 In the profile tree, click **Shortcuts** and then click **Add**.
- 2 In the **Add Shortcuts** dialog, select the shortcut files you want to add and click **Add**.

You can use variables such as **Common Start Menu** to represent the location of the shortcut. In the **Variables** list, click a variable. Click **Add** to add the shortcut files.

Copying Files

A profile can include a list of files to retrieve during an audit. These files can be text files or binary files.

Unless you use a variable to specify the exact location of the file to copy, Triage copies all occurrences of the file it finds on the computer. Therefore, it is strongly recommended to use variables when specifying files to copy.

Attaching Copied Files

Binary files are always attached to audit reports. And by default, ASCII files are also attached to audit reports (so the audit report contains only a reference to the copied files, which are stored externally in the file system).

Attaching the copied files reduces the size of the audit report and reduces the amount of time required to load the report into Triage Console. It also allows you to use the application associated with the file type to open or edit the file.

You can include copied ASCII files in the audit report file by setting the **Attach Copied Files** attribute to **False**. Including copied files in an audit allows you to automatically compare them when you compare audit reports.

However, including copied files increases the size of the audit report and the time required to load the report into Triage Console. It also means you cannot open the file in another application (such as Notepad).

If you are retrieving copies of large files, attach them to the audit report.

Auditing with Windows Management Instrumentation

Windows Management Instrumentation (WMI) is the Microsoft implementation of Web-Based Enterprise Management (WBEM), which is an industry initiative to develop a standard technology for accessing management information. Such management information includes information on the state of system memory, networks, devices, and other information on client status. WMI offers a powerful set of base services that include query-based information retrieval and event notification.

WMI is supported on Windows 2000, XP, and Me, and available as an optional install for Windows 95 OSR 2, 98, and NT4 SP5.

WMI Components

An application profile can specify a list of WMI components and their properties to audit. To customize the WMI information audited, you can:

- View the properties and change their values.
- Reload the factory default settings for a category or an object.

To customize the WMI Components list, you can:

- Remove a component or a category from the list.
- Add additional WMI Components for selection.

Editing WMI Category Properties

Display name Caption of the WMI category (referred to as a namespace).

Namespace Specifies the server path of the namespace.

Editing WMI Component Properties

Display name Caption of the WMI component.

Query Associators If **True**, Triage audits all associated WMI objects.

WMI SQL SQL statement that specifies what information to retrieve. You can change the name and the value of the WMI SQL property. You can also add new SQL statements for the same object.

For example, to query the NT event log for errors only and separate the result of each query under three different categories: Application Errors, Security Errors and System Errors:

- 1 Rename the default WMI SQL to "Application Errors" and modify the SQL statement to:

```
SELECT * FROM win32_NTLogEvent WHERE
LogFile = "Application" AND Type = "Error"
```

- 2 Add a WMI SQL property and rename it to "Security Errors". Set the WMI SQL statement to:

```
SELECT * FROM win32_NTLogEvent WHERE
LogFile = "Security" AND Type = "Error"
```

- 3 Add a WMI SQL property and rename it to "System Errors". Set the WMI SQL statement to:

```
SELECT * FROM win32_NTLogEvent WHERE
LogFile = "System" AND Type = "Error"
```

For Triage to audit a WMI Component and return information about the component, you must provide at least one WMI SQL property for the component.

Customizing the WMI Components List

To audit a WMI object not listed in the factory default list, you use the Customize feature to first add it to the list.

To add a new component to the list:

- 1 Click **Customize**.
- 2 In the **Customize** dialog, click the check box for component you want to add.
- 3 Click **Add**.

You can change the display name of the object to a more user-friendly name by entering the new name in the Display Name column. (Objects prefixed with a "", are objects containing a modified Display Name.)*

To add several objects at once, hold down the CTRL key and then click each object you want to select. Hold down the SHIFT key to select a range of files. Click Add to add the selected objects.

To add a new category you must edit the **UserWMI.INI** file and add it under the [Namespaces] section.

To remove a component or category from the list:

Click a WMI component or category and then click Remove. You cannot remove any of the factory default WMI Objects from the list.

WMI INI File Format

The list of WMI categories and components displayed in Triage is defined by the **MqWMI.INI** and **UserWMI.INI** files.

- **MqWMI.INI** provides the list of default WMI categories and their components. Settings in the **MqWMI.INI** are referred to as factory settings and cannot be removed using Triage.
- **UserWMI.INI** contains the categories and components added using Triage.

If you edit the INI files manually, you must follow to the file format described below so that Triage can load these files. Categories (namespaces) must be added manually to the **UserWMI.INI** file following the format outlined below. To add a category, you must add an entry under the **[Namespaces]** section:

Category (Namespace) entry:

```
[Namespaces]namespace=type:
display name:namespace server path
where type can have two possible values:
```

- 0 (Default)
- 1 (Custom)

For example:

```
[Namespaces]CIMV2=0:win32 Environment:
\\.\root\cimv2
```

To add a category's component list, you must add object (class) entries under its corresponding namespace section:

Component (Class) entry:

```
[namespace]class name=type:
displayname:SQLstatement
```

For example:

```
[CIMV2]win32_DMACHannel=0:DMA Channel:SELECT
* FROM
win32_DMACHannelwin32_IRQResource=0:IRQ
Resources:SELECT * FROM
win32_IRQResourceStoppedManualServices=1:Sto
pped Manual
Services:SELECT * FROM win32_Service WHERE
StartMode = "Manual" AND State = "Stopped"
```

StoppedManualServices is an example of a custom class that you can add that adheres to the format guidelines.

Auditing Database Information

The Triage/db install option extends the auditing capabilities of Triage to include database configuration information and database content. Triage/db can collect information for any ODBC-compliant database such as Oracle, Microsoft SQL Server, and Microsoft Access.

Due to the nature of how ODBC is implemented, (multiple layers of programs and drivers communicating with each other), troubleshooting can be a challenge.

With Triage/db, when an ODBC call fails, you no longer need to spend hours trying to determine whether it is a problem with client libraries, or a net protocol mismatch, or even a database engine not running, Triage/db can collect all the information required to perform a proper diagnosis in minutes.

ODBC Database Configuration

Triage steps you through the process of specifying what to collect about a user's ODBC installation. The ODBC configuration information is grouped into categories:

System DSNs Data Source Name, registry security, description, system database, ODBC driver, User, DSN configuration settings, and more.

User DSNs Data Source Name, registry security, description, system database, ODBC driver, User, DSN configuration settings, and more.

File DSNs Data Source Name, and file information (location, size, attributes).

ODBC Drivers File version information (file name, location, file version, attributes, and more), API level, ODBC driver version, SQL level, and more.

Database Connection Information

Triage/db can retrieve data from any database table a user has access to read. The Database Connection Editor provides three ways to retrieve data from a database:

By selecting tables Triage returns the content of the selected tables.

By selecting stored procedures Triage returns the result of running the stored procedure.

By specifying an SQL statement Triage returns the result of running the SQL statement.

With Database Connection Information, you can add new connections and edit or remove existing connections.

To add a database connection:

- 1 In the Database Connections dialog, click **Add**.

The Database Connection Editor opens to allow you to create a new Database Connection.

- 2 In the **Type** list, click the type of connection.
- 3 For an ODBC connection, click **Browse** and then click the type of **DSN**.

User DSN Click a user DSN and click OK.

System DSN Click a system DSN (if any) and click OK.

File DSN In the Look in Drive list click a drive, then click a file DSN and click OK.

SQL Server Enter the names of the SQL server and the database, a user ID and password, and then click OK. You can use variables in any of the fields.

No DSN Enter a connection string that will open the database. For example:

```
DRIVER=SQL Server;SERVER=YourServer;
UID=YourLogonName;PWD=YourPassword;
APP=Microsoft®Access;WSID=YOURMACHINE;DATABASE=YOURDATABASE)
```

You can use variables in the connection string.

Selecting Data to Collect

You can select the data to be collected from the connection as Tables, Procedures and SQL Statements.

To select tables:

- To select all the tables in the DSN, click the check box beside **ODBC Tables** in the list of tables.
- To select only certain tables in the DSN, click the check boxes beside the tables you want to include.

To select procedures:

- To select all the procedures in the DSN, click the check box beside **ODBC Procedures** in the list of procedures.
- To select only certain procedures in the DSN, click the check boxes beside the procedures you want to include.

To enter SQL statements:

- 1 Under **SQL Statements**, click **Add** to add a query to your connection.

- 2 Under **Edit SQL Statement**, type a name and SQL statement (for example: Select * from tblAttachments).
- 3 Click **Test** to view the results of your query in your default Web browser.
- 4 When you are satisfied with the query, click **Apply**.

You can add more SQL Statements to your connection, and edit or remove existing ones.

You can use variables in the SQL statement.

Collecting Diagnostics for IIS

You can collect information about the Web sites, virtual directories, FTP sites, and SMTP servers on an IIS Web server.

To collect IIS diagnostics:

- 1 In the profile tree (Details view), click **Audit Preferences**.
- 2 In the **Value** list of the **Internet Information Server** attribute, click **True**.

Collecting Security Information

You can collect files, shares, and registry security information.

To collect security information:

- 1 In the profile tree (Details view), under **Audit Preferences**, click **Security**.

- 2 Set the **Include File Security, Include Registry Security, or Include Share Security** attribute to **True**.

Collecting User Information

Triage Client includes a User Information dialog where users can enter information about themselves and the product.

You can customize this dialog by:

- Adding and removing fields. There is a maximum of eight fields.
- Changing field captions.
- Making fields required. Required fields must be filled in by a user before sending the audit report.
- Changing the order of the fields on the dialog.
- Defining a list of possible values by entering a semi-colon separated list of values. For example, "Triage;Triage Pro;Triage/db".

In Triage Client, the list of values are displayed in a drop-down list box.

Triage Client also includes a **Problem Description** dialog, which you can hide, or make a required field.

The Request User Info attribute controls whether Triage Client (Classic mode) displays the User Information and Problem Description dialogs.

Chapter 5: Adding Self-Repair Capabilities

A profile can include repair rules that specify how to detect and fix problems. For example, a repair rule for a file looks like this:

```
If Audit Status = Found AND
    Size (bytes) = 987,136
    Do Nothing
Else
    Fix it
```

Triage Client evaluates the self-repair rules when it audits a computer, and notifies the user of any problems. The user can then fix the problem, test the fix, and if necessary, undo the fix.

The general form of a repair rule looks like this:

```
if ( condition )
    action1
else
    action2
```

condition is a logical expression that tests the values in an audit report.

actions are predefined actions such as Fix it, Display Message, and Do Nothing. Fix it depends on the type of object.

Setting Up Self-Repair

To set up self-repair, you add self-repair rules to the profile, and then create a self-repair package that contains the files needed to fix problems with files and ActiveX controls. Like profiling, you can do this all automatically for all items in the profile, or manually for specific items.

To set up self-repair:

- 1 Generate default self-repair rules for the different items included in a profile. You can define self-repair rules for files, ActiveX controls, shortcuts, environment variables, and registry entries.
- 2 If necessary, edit the generated self-repair rules.
- 3 If necessary, enable (or disable) the self-repair rules for individual items. By default, generated self-repair rules are always enabled.

To enable and disable rules, select one or more objects, click **Self-Repair**, and click **Enable Conditions** or **Disable Conditions**.

These commands are shortcuts for setting the **Enable Self-Repair** self-repair preference.

- 4 Enable self-repair in Triage Client.


In the profile tree, click **Self-Repair** under **Audit Preferences**. Set **Enable Self-Repair** to **True**. This allows Triage Client to apply the self-repair rules to diagnose and fix problems.
- 5 Set the **Run Self-Repair** profile attribute.

If **Run Self-Repair** is **True**, Triage Client automatically pops up the Self-Repair dialog when it finds a problem.
- 6 If you want users to be able to test fixes by running your application, set the **Executable Name** preference to the name of your application executable.
- 7 Save the profile.
- 8 Copy the self-repair package to your Web (or ftp) site.

Generating Repair Rules

You can generate repair rules for files, ActiveX controls, shortcuts, environment variables, and registry values in the profile.

To generate repair rules for all items

- 1 In the console tree, click a profile.
- 2 In the Details view toolbar, click .
- 3 Click **Yes** to perform an audit of the local computer. Triage Console uses the information gathered during the audit to build the repair conditions.
- 4 Type a URL for the self-repair package. This is location from which Triage Client downloads the package. For example, "http://www.server.com/app-repairs.zip". You need to include a file name. Triage Console will use this file name for the package, which it saves in the same location as the profile.

After you update all the self-repair rules, you can edit and customize individual self-repair rules, and then regenerate the self-repair package. But you must lock any customized self-repair rules so they are not overwritten with default, generated rules.

To generate repair rules for specific items:

- 1 In the profile tree, click **Files, Registry, ActiveX Controls, Environment Variables, or Shortcuts**.
- 2 Select one or more items.

Use the Shift and Ctrl keys to select multiple objects, or drag the pointer over the objects you want to select. To select by dragging, point to a blank area (for example, the whitespace after an item name) and then drag the bounding outline.

- 3 Click **Self Repair** and then click **Auto Build**.

Triage Console generates default repair rules for the selected objects.

Editing Repair Rules

To edit a repair rule:

- 1 In the profile tree, click **Files, Registry, ActiveX Controls, Environment Variables, or Shortcuts**.
- 2 Click an item (a file, ActiveX control, shortcut, environment variable, or registry entry).
- 3 Click **Self Repair** and click **Build Condition**.

Defining Conditions

A condition is one or more expressions joined by And or Or. Each expression tests the value of an object property. For example:

```
Audit Status = Found AND  
Size (bytes) = 987,136
```


To define a condition:

- 1 Click **Add**.
- 2 Click in the **Property** box and select an object property. The **Property** box lists the object properties that can be used to build a condition.

Use the **Audit Status** property to test whether the object was found during the audit.

Don't use the **Location** property to define a condition. Instead, set the Target Directory attribute.
- 3 Click in the **Test** box and select a logical test.
- 4 Click in the **Value** box.

The value you enter here is compared against the value in an audit report.

Click **Get**  to get the current value of a property.

To test environment variables like PATH Use the Contains test operator instead of the = operator. When Triage gets the current value of the PATH environment variable, it gets the value for the current process (Triage Console). So the path to the Triage installation directory is added to the start of the PATH variable.

Defining Actions

Action Arguments

The Display Message and Go to URL actions each have an argument. For Display Message, the argument is the text to display. For Go to URL, the argument is the URL.

Action Captions

The caption is the text displayed on the button beside a problem in the Details view. The default caption is “Fix it”.

The width of the button is controlled by the Action column width preference. See “Configuring Self-Repair” on page 38.

Delete it

Deletes registry entries.

Delete it for Registry Keys Triage can delete a registry key and all of its descendants.

You do not need to add any condition to operate on keys. If the key exists, Triage considers that the condition is met. If the key does not exist, the condition is not met.

Delete it for Registry Values If you want to delete the registry value regardless of its current value, do not specify any condition. If the value exists, Triage will delete it.

Display Message

Displays the message specified by the **Argument** field.

```
if ( condition )
    Display Message
    Argument = "Condition met!"
else
    Display Message
    Argument = "condition failed!"
```

Do Nothing

No action.

Fix it

Fix it for Files **Fix it** for files either copies or downloads a good version of the file to the specified target directory. Triage Client first scans the user's system for a file that matches the specified condition. If it finds one, it copies the file to the specified target directory. If Triage Client does not find one, it tries to download the file from the web:

- If the **Self-Repair Package** attribute is set, Triage Client downloads the file to the target directory.

- If the **Self-Repair Package** attribute is not set, Triage Client pops up a file browser so the user can locate a good version of the file (for example, on the product CD). The specified file is then copied to the target directory.

Note that an existing file is renamed when the good version of the file is copied to the target directory.

The target directory is specified by the **Target Directory** attribute. If **Target Directory** is not set, Triage Client pops up a file browser so the user can choose a target directory.

If Triage Client finds both good and bad versions of a file on a computer, it checks:

- Is there a version of the file in memory?
- Is there a version of the file in the user's path or in the target directory?

If Triage Client finds that a bad version of the file is being used by the application, it copies a good version of the file to the target directory.

If there is no file in memory or in the target directory, and at least one good version of the file was found, then Triage Client assumes everything is ok.

Fix it for ActiveX Controls and Self-Registered Files If the file is not registered, is the wrong version, or is missing, Triage gets the file from the self-repair package and registers it. If the file is already present on the computer but is just not registered, Triage registers it.

Fix it for Shortcuts If a shortcut is broken, Triage tries to fix it based on the path specified in the condition. But if the path does not point to an existing file, Triage scans the system for the first occurrence of a file with the same name and fixes the shortcut to point to that file.

Fix it for Registry values If a registry value does not meet the specified condition, Triage updates the registry entry according to the criteria specified in the condition. Triage can repair individual registry values only, not complete hierarchies.

Fix it for Environment Variables Fix it updates the value of the environment variables to match the value found when the application was profiled.

Go to URL

Starts the default browser and loads the URL specified in the **Argument** field.

Rename it

Renames a file.

Unregister it

Unregisters an ActiveX control.

Setting Attribute Values

To edit the attributes of a repair rule:

- 1 In the profile tree, click **Files, Registry, ActiveX Controls, Environment Variables, or Shortcuts**.
- 2 Click an item (a file, ActiveX control, shortcut, environment variable, or registry entry).
- 3 Click **Self Repair** and click **Build Condition**.
- 4 In the **Attributes** list, click in the **Value** field to edit the attribute value.

Auto Execute Action

If **True**, Triage automatically executes the specified repair action.

Description

Text displayed between the problem title and the Show Details section in the **Self-Repair Diagnosis** dialog.

Problem Description



Enable Self-Repair

If **True**, Triage applies the repair rule. If **False**, the rule is disabled.

Locked

If **True**, the self-repair rule is not updated when you update all self-repair rules. When all self-repair rules are updated, all unlocked self-repair rules are replaced with generated self-repair rules. If you have customized self-repair rules, you should lock them.

Problem Priority

By default, Triage sorts problems by priority, with the highest priority problems appearing at the top of the list. Lower numbers indicate higher priority.

Self-Repair Package


Http address, ftp address, UNC path, or local path of file to download. Can be an individual file or a ZIP file.

Target Directory

Specifies where to put a file on the user's machine when the problem is fixed. Used when a file is downloaded from Internet Location or when the user locates the file with the File Browser dialog.

Target Directory is typically set to the value of a variable such as `$(WinDir)` or `$(ApplInstallDir)`.

Title

Text displayed after the **Problem:** label for a problem. To type or edit a multi-line title, click .

Enabling and Disabling Repair Rules

To enable and disable self-repair rules

- 1 In the profile tree, click **Files, Registry, ActiveX Controls, Environment Variables, or Shortcuts**.
- 2 Select one or more items.
Use the Shift and Ctrl keys to select multiple objects, or drag the pointer over the objects you want to select. To select by dragging, point to a blank area (for example, the whitespace after an item name) and then drag the bounding outline.
- 3 Click **Self-Repair** and click **Enable Conditions** or **Disable Conditions**.

These commands are shortcuts for setting the **Enable Self-Repair** self-repair attribute.

Locking Customized Repair Rules

If you customize any repair rule conditions, you must lock the repair rules so the conditions are not overwritten with the defaults generated when you update all repair rules.

The condition is the logical expression tested by the if...then statement.

Creating Self-Repair Packages

A self-repair package is a ZIP file that contains the files and ActiveX controls required to fix problems. Triage Client automatically downloads the self-repair package and extracts the required files when a user applies a fix. The self-repair package is automatically removed from the user's system when Triage Client exits.

When you use the **Update all self-repair rules and create self-repair package** feature, Triage automatically builds the self-repair package and puts it in the same directory as the profile (the .trp file).

When you create your own self-repair rules, you must create the self-repair package yourself. You can use any compression tool that supports the ZIP format.

When you use **Auto Build** to generate self-repair rules for individual items, you can set the default self-repair package for new rules.

To set the default self-repair package:

- 1 Click **Self-Repair** and then click **Self-Repair Package**.
- 2 Type the path to the ZIP file. The path can be an http address, ftp address, UNC path, or local path.
- 3 Click **OK**.

To set the self-repair package for items that already have self-repair rules, click the objects, click Self-Repair, and click Set Self-Repair Package.

To clear the self-repair package for an item, click the object, click Self-Repair, and then click Build Condition. In the Attributes list, delete the value of the Self-Repair Package attribute.

Configuring Self-Repair

The Self-Repair preferences allow you to configure the self-repair capabilities of Triage Client.

Diagnosis Show Details If **True**, Triage displays problem details.

Enable Self-Repair If **True**, Triage Client uses the self-repair information in the profile to diagnose and fix problems. Otherwise, Triage Client ignores all self-repair information in the profile.

Note that self-repair can be disabled for individual objects (such as files, shortcuts, and ActiveX controls).

Executable Name The executable that Triage Client tries to run when the user tests the fixes made by Self Repair.

Run Self-Repair (Classic mode only)

If **True**, Triage Client automatically pops up the Self Repair dialog after the audit if any problems were found. Otherwise, the user must click the Diagnose and fix button to run Self Repair.

Update Profile If **True**, Triage Client updates the self-repair rules in the profile and creates a self-repair package.

If **False**, the /updateprofile command-line switch determines if Triage Client updates the self-repair rules in the profile and creates a self-repair package

Chapter 6: Configuring Triage Client

Interface Modes

The Triage Client interface mode is controlled by the **/mode** command-line option. If you are packaging Triage Client for eSupport, you can add **/mode** option to the generated command line by setting the **Web Client GUI Mode** audit preference.

Classic Mode

In Classic mode, all actions (such as auditing, saving, and sending) are performed from main dialog. Users can switch to Wizard mode by clicking Wizard at the bottom of the dialog.

Classic Mode



In ClassicOnly mode, users cannot switch to Wizard mode.

Wizard Mode

In Wizard mode, the Next (and Back) buttons step the user through a fixed sequence of actions. Users can switch to Classic mode by clicking Classic.

Wizard Mode



In WizardOnly mode, users cannot switch to Classic mode.

Configuring Classic Mode

In Classic mode, Triage Client consists of a main dialog with several secondary dialogs.

License Agreement Dialog, Splash Screen

To customize the license agreement text, edit the file **license.txt** in the Triage installation folder.

Use **/nologo="1"** to skip the license agreement and splash screen.

Welcome Dialog

To skip this dialog, use the **/profile** command-line argument. Note that when Triage Client is packaged for eSupport, it always uses **/profile**, so the user will never see this dialog.

To remove the **Wizard** button, set **Web Client GUI Mode** to **WizardOnly** (or if you are shipping Triage Client on CD, you'll have to use the **/mode** command-line argument).

Main Dialog

To automatically start the audit, use the **/autostar-audit** command-line argument. If used with the **/profile** argument (for example, in an eSupport package), the audit starts immediately after the splash screen is removed.

To automatically exit after the audit is finished, use the **/autoexit** command-line argument.

To remove the **Wizard** button, set **Web Client GUI Mode** to **WizardOnly**. If you ship Triage Client on CD, you'll have to use the **/mode** command line argument.

View Audit Report Dialog

The **View and edit audit report** button opens the **View Audit Report** dialog box.

To automatically display the **View Audit Report** dialog, set **Show Audit Results** to **True**.

To remove the **Delete** button from the **Audit Viewer**, set **Enable Audit Delete** to **False**.

Options Dialog

To prompt the user to fill in the **User Information** and **Problem Description** dialogs, use the **Request User Info** attribute.

You can customize the fields that appear on the **User Information** tab.

To remove the **Problem Description** dialog, click the profile in the console tree, click **User Information**, and then under **Problem Description**, clear the **Visible** check box.

Self-Repair Dialog

To enable the **Diagnose and fix the problem** button, set **Enable Self-Repair** to **True**.

To automatically apply the self-repair rules, set **Run Self-Repair** to **True**. By default, the user must click the **Diagnose and fix the problem** button to run self-repair.

Configuring Wizard Mode

In Wizard mode, Triage Client consists of this sequence of dialog boxes.

License Agreement Dialog and Splash Screen

To customize the license agreement text, edit the file **license.txt** in the Triage installation folder.

Use **/nologo="1"** to skip the license agreement and splash screen.

Welcome Dialog

To customize the text on this dialog, create a file named **TriageIntro.txt** in the Triage installation folder. Put the text you want to see on the **Welcome** dialog in this file. Add the file to **cab.txt**. This works for Wizard mode only.

To remove the Classic button, set **Web Client GUI Mode** to **WizardOnly**. If you are shipping Triage Client on CD, you'll have to use the **/mode** command line argument.

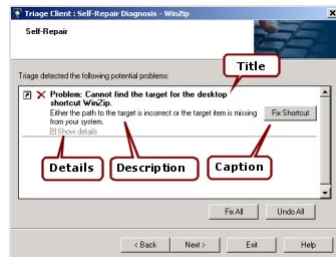
Profile Info Dialog

To skip this dialog, use the **/profile** command-line argument. Note that when Triage Client is packaged for eSupport, it always uses **/profile**, so the user will never see this dialog.

Self-Repair Diagnosis Dialog

To enable self-repair and display the Self-Repair Diagnosis dialog, set **Enable Self-Repair** to **True**.

Elements of Self-Repair Diagnosis



To hide problem details for all problems, set the **Diagnosis Show Details** self-repair preference to **False**.

To customize the text displayed for a problem, use the **Title** and **Description** repair rule attributes.

To sort problems, assign priorities to each problem with the **Problem Priority** repair rule attribute. Triage Client will display the problems sorted in order of increasing priority.

To change the caption of the **Fix It** button, set the **Caption** of the repair condition. The width of the button is controlled by the **Action column width** preference.

Self-Repair Results Dialog

The **Test** button runs the executable specified by the **Executable Name** self-repair preference.

User Information Dialog

To add, edit, or remove user information fields, click the profile in the console tree, and then click **User Information**.

Problem Description Dialog

To remove the **Problem Description** dialog, click the profile in the console tree, click **User Information**, and then under **Problem Description**, clear the **Visible** check box.

Send Audit Report Dialog

- The **View** button opens the **View Audit Report** dialog. To remove the **Delete** button from the **Audit Viewer**, set **Enable Audit Delete** to **False**.
- To automatically display the **View Audit Report** dialog when the user clicks **Next** on the preceding dialog, set **Show Audit Results** to **True**.
- To remove the **Mail Options** button, set **Enable Mail Options** to **False**.
- To customize the message displayed when Triage Client is unable to send the audit report, use the **Send Error Message** preference.

Finished Dialog

To remove the **Save** button, which allows the user to save the audit report, set **Enable Save Report** to **False**.

To automatically save the audit report, set **Auto Save** to **True**.

To customize the text displayed on the **Finished** dialog, use the **Finished Message** preference.

Command-Line Arguments

/autoexit

```
/autoexit="0|1"
```

/autoexit="1" automatically exits Triage Client when the audit is finished.

/autoexit works only with Classic mode (however, the implied **/autoexit** that comes with **/hidden** works in both Classic and Wizard mode).

/autoexit is ignored if the profile includes self-repair rules.

/automail

```
/automail="0|1"
```

/automail="1" automatically e-mails the audit report when the audit is completed. If Triage Client is not in hidden mode, it will prompt the user to send the audit report.

/autostartaudit

```
/autostartaudit="0|1"
```

Automatically starts the audit after the profile is loaded. Applies to Classic mode only. In Wizard mode, the user must always click Next or Start Audit.

/copyfiles

```
/copyfiles=filepath[,filepath]...
```

Adds files to the list of files to copy. For example:

```
/copyfiles="C:\winnt\system.ini,C:\winnt\win.ini"
```

/hidden

```
/hidden="0|1"
```

/hidden="1" runs Triage Client minimized, and is equivalent to /hidden="1" /autostartaudit="1" /autoexit="1".

Requires **/profile**, because otherwise Triage Client must be displayed so the user can load a profile.

The License Agreement and Splash Screen are displayed unless you use the /nologo="1" option.

For example, to run a "hidden" audit and automatically send the audit report, use /hidden with /automail:

```
TriageClient.exe /nologo="1" /hidden="1"  
/profile="aProfile.trp" /automail="1"  
/mail="stblair@metaquest.com"
```

/mail

Sets the e-mail recipient of the audit report. Overrides the value of the E-mail Address profile attribute.

/mode

```
/mode="Classic | ClassicOnly | wizard  
| wizardOnly"
```

Sets the interface mode of Triage Client. Triage Client has two basic modes: Wizard and Classic. Wizard mode guides the user through a series of steps, while Classic mode provides a single main dialog that allows the user to perform actions in any order. Users can switch between the two modes when Triage Client first starts, unless you use the ClassicOnly or WizardOnly modes.

/nologo

```
/nologo="0|1"
```

/nologo="1" skips the splash screen and the license agreement.

/profile

Loads a profile (a .trp file). For example, the following command starts Triage Client and generates an audit report for myapp:

```
TriageClient.exe /profile="myapp.trp"
```

If you put this command in a script file or a shortcut, users don't have to worry about loading the correct profile. All they have to do is edit, save, and send the generated audit report.

/profile can be a local path, a UNC path, or a URL.

/props

Sets profile properties.

Profile properties are specified in a .trp file, which is an XML document. For example, in a .trp file, the Internet Information Server, Auto Exit, and Auto Save profile attributes are specified by the following XML:

```
<HeaderInfo Name="Audit Preferences">  
...  
<Internet_Information_Server value="0" />  
<Auto_Exit Value="0" />  
<Auto_Save Value="" />  
...  
</HeaderInfo>
```

The syntax for setting these attributes from the command-line is:

```
/props="Audit Preferences.Internet  
Information Server=1;Audit Preferences.Auto  
Exit=1;Audit Preferences.Auto  
Save=c:\foo.cab"
```

Note that underscores in the .trp file must be replaced with spaces on the command line.

For boolean properties, such as Auto Exit, True corresponds to any non-zero value (1 or -1), and False corresponds to zero (0).

Here's another example:

```
<HeaderInfo Name="Audit Preferences">  
...  
<SelfRepairHeader Name="Self-Repair">  
...  
<Enable_Self-Repair value="0" />  
...  
</SelfRepairHeader>  
</HeaderInfo>
```

The syntax for setting the Enable Self Repair attribute from the command-line is:

```
/props="Audit Preferences.Self-Repair.Enable  
Self-Repair=1"
```

/save

Saves the audit report (.tra) to the specified path.

/smtp

Specifies the SMTP server to use when sending SMTP mail.

For example, if you want your customers to send audit reports with Internet mail without having to specify a SMTP server, use the /smtp option to specify your SMTP server:

```
TriageClient.exe /profile="myapp.trp"  
/save="myapp" /mail="me@address.com"  
/smtp="smtp.hip.cam.org"
```

/updateprofile

Updates the self-repair rules and creates a self-repair package. This allows users to take a "snapshot" of their system, which they can use to repair any problems that occur later.

```
TriageClient.exe  
/profile="someapp.trp"  
/updateprofile="1"
```

After Triage Client finishes the audit, it updates the profile and saves the self-repair package (a .ZIP file). The .ZIP file is saved in the same folder as the profile (.TRP) file.

If the value of /profile is a URL, Triage Client does not update the self-repair rules or create the self-repair package (because it cannot save the updated profile or the self-repair package to the specified URL).

If you set the **Update Profile** self-repair preference to True, Triage Client updates the self-repair rules and creates a self-repair package every time it is run.

Web Publishing Audit Reports

Web Publishing allows Triage Client to post audit reports to your server via the HTTP or the FTP protocols.

Instead of sending an e-mail message with an attachment containing the audit report, Web Publishing can upload the audit report to a customizable location on your server.

All that is required is to specify the http or ftp location of your server and if required, a user name and a password. This information is saved in the profile. Note that the password is encrypted.

Triage supports a generic Web Publishing mechanism (referred to as Triage Web Publishing) as well as Microsoft Web Publishing.

Microsoft Web Publishing can automatically post to a variety of Web servers.

The naming convention for the published file is as follows:

```
<E-mail address>-<Product Name>-<Date>-<Time>
```

For example:

```
support@metaquest.com-Triage-2000-08-31-10-  
13-072.cab
```

Web Publishing Attributes

Web Publishing attributes are options for controlling how to upload audit reports to your web server.

E-mail Notification Address After a file is successfully uploaded to your server, an e-mail can be sent to the address specified in this field to notify your support staff that a new request has arrived.

Http or Ftp Address Http or Ftp address where the file should be uploaded.

Invoke Servlet Http address of servlet to invoke. The servlet is invoked after the audit report is posted.

You can use the \$(Audit Report Filename) variable to reference the audit report. For example:

```
http://www.metaquest.com/MyServlet&AuditReport=$(AuditReportFileName)
```

Password Password needed if the specified Http Address requires authentication.

Status Field Name

Use Microsoft Web Publishing If true, Triage will use the Microsoft Web Publishing SDK to post files to the server. This option is recommended if you are running an NT server with Internet Information Server (IIS).

User Name Needed if the specified Http Address requires authentication.

Verb Depending on your web server configuration, Triage can upload audit reports using HTTP POST or HTTP PUT. If you are uploading to an FTP server, use FTP PUT.

Client Requirements

The components required to publish files are typically part of the operating system, so there should be no need to install any additional software.

Server Requirements

If you decide to use Triage Web Publishing, you will need to configure your server to accept uploads. Because there are numerous web servers available, we cannot provide detailed

instructions on how to configure each server. Please contact your system administrator or your web server vendor for additional help.

If you use Microsoft Web Publishing, the server requirements are described below.

If you are using Microsoft Internet Information Server as your web server you need to install the Microsoft Posting Acceptor on your server.

Microsoft Posting Acceptor allows Microsoft Internet Information Server (IIS) to accept Web content posts (files) from Microsoft Web Publishing or other clients using the RFC1867 multi-form/posting method through an http connection.

If Microsoft Web Publishing is unable to post to your site, you may want to consider including a meta tag in your default page and adding a customized postinfo (postinfo.ini) file to fit the configuration of your site.

A sample postinfo.ini file is provided in your Triage installation directory.

Copy the postinfo.ini to your server and add the following line to your default page (typically index.htm):

```
<META name="postinfo"
content="/postinfo.ini">
```

There are no specific requirements to upload to an FTP server.

Chapter 7: Setting Up Triage eSupport

Triage eSupport allows you to provide customers with automated problem diagnosis and repair, all from your technical support Web.

To do this, you use Triage Console to generate the files you need to add Triage eSupport to your Web site:

- An eSupport package (a JAR file) that is a compressed archive of all the files (such as the Triage Client executable and the application profile) that need to be downloaded to audit a user's machine.
- A set of static HTML pages that you add to your Web site so that users can download the eSupport package.
- A Java applet that handles the download and execution of Triage Client.

When a user downloads the eSupport package, the Java applet extracts the package contents into a temporary directory and runs Triage Client. After Triage Client finishes the audit, it sends the generated audit report back to you. Triage Client is then removed from the user's system.

To set up Triage eSupport

- 1 Get a digital certificate so you can digitally sign the eSupport Package.
- 2 Customize the eSupport HTML pages (the HTML pages that allow users to get the eSupport Packages).
- 3 Set the eSupport audit preferences that control the Triage Client command line.
- 4 Generate the eSupport package.
- 5 Add the download pages to your Web site.

Signing

You must digitally sign the eSupport Package for Netscape Communicator. Signing the eSupport Package is optional for Internet Explorer. Note that you need separate certificates for Netscape and Internet Explorer.

Digital signing allows users to get reliable information about the objects they download. Users can identify who published a software component and verify that no one tampered with it before downloading it from the Internet.

To sign an eSupport package:

- 1 Get an object-signing certificate. You can purchase object-signing certificates from VeriSign, BelSign, or Thawte Certification.
- 2 Once you have your certificates, install the Netscape certificate.
- 3 Fill in the IE Digital Signing and Netscape Digital Signing dialog boxes.

Triage automatically signs the eSupport packages when it generates them.

In the Netscape Object Signing dialog box, the Location of certificate directory is typically a subdirectory of Netscape\Users (for example, c:\Program Files\Netscape\Users\stblair).

To install a digital certificate for Netscape Communicator:

- 1 Start Netscape Communicator.
- 2 In the **Communicator** menu, click **Tools** and then click **Security Info**.

- 3 In the **Security Info** dialog, select **Certificates/Yours**.

This lists all the certificates you have on your machine.

- 4 Select **Import a Certificate**.
- 5 Enter your password.
- 6 Enter the name of your private key (a .p12 file).

This procedure registers your certificate on your computer. The certificate is stored in a database located in your Netscape\Users directory (for example, d:\Program Files\Netscape\Users\rakhras).

Customizing eSupport Web Pages

When Triage Console packages Triage for eSupport, it generates HTML pages for downloading the package. You can find the templates for these HTML pages in the Triage installation directory. You can modify these templates (or the generated pages) to provide the same look and feel as your other Web site pages.

TriageDownload.htm A plain HTML file that provides some introductory information to the user, and a button that redirects to the **TriageGo.htm** page.

esupport.htm Template for **TriageGo.htm**. Contains an embedded Java applet (the MqSmartUpdate applet). The MqSmartUpdate applet downloads the eSupport package and then automatically extracts and runs Triage Client. The **TriageGo.htm** page works for all versions of Internet Explorer and Netscape.

TriageHelp.htm Help and FAQs for Triage Client.

msiasecurity.htm Help and instructions for changing the security settings in Internet Explorer.

TriageDownload.htm

TriageDownload.htm defines a **runTriage()** function that redirects the user to **TriageGo.htm**. If you change the file name of **TriageGo.htm**, you need to update the **TriageGo()** function.

```
<SCRIPT LANGUAGE="JavaScript">
function runTriage () {
    window.location.replace('TriageGo.htm');
}
</SCRIPT>
```

A button calls runTriage():

```
<form name="runTriageForm">
<center>
  <input
    type="button"
    name="runTriageButton"
    onClick="runTriage();"
    value="Begin! »"
  >
</center>
</form>
```

esupport.htm

esupport.htm is the template for **TriageGo.htm**. **esupport.htm** contains an `<applet>` element that embeds the MqSmartUpdate java applet in the page. When you package Triage Client for eSupport, Triage updates the command parameter of the applet. The command parameter specifies the command-line arguments for Triage Client.

Do not edit the command parameter element in **esupport.htm**. Triage will overwrite your changes when it creates the eSupport package.

To customize the Triage Client command-line, use the eSupport audit preferences (see “eSupport Preferences” on page 58).

```
<APPLET
  CODE
  ='com.metaquest.smartupdate.MqSmartUpdateApp
  let.class'
  ARCHIVE ='MqSmartUpdate.jar'
  CODEBASE='.'
  HEIGHT  = 50
  WIDTH   = 400
  BORDER  = 0
  NAME    = MqSmartUpdateApplet MAYSCRIPT>
  <PARAM name=downloadURL
value='netscape.jar'>
  <PARAM name=downloadMsg value="Downloading
Triage Client. Please wait...">

  <PARAM name=downloadCallback value="">
  <PARAM name=inflate value="1">
  <PARAM name=targetDir value="Triage">
  <PARAM name=mode value="1">
  <PARAM name=command
value="TriageClient.exe">
  <PARAM name=commandMsg value="Running
Triage...">
  <PARAM name=commandCallback value="">
  <PARAM name=yesNoCallback value="">
  <PARAM name=background value='FFFFFF'>
  <PARAM name=foreground value='000000'>
  <PARAM name=progressBar value='1'>
  <PARAM name=progressBarBackground
value='FFFFFF'>
  <PARAM name=progressBarForeground
value='808080'>
  <PARAM name=CABBASE
value="MqSmartUpdate.cab">
</APPLET>
```

The MqSmartUpdate applet displays a progress bar while downloading and running Triage Client. You can customize the progress bar by setting the applet parameters:

downloadMsg Message displayed while downloading the eSupport package.

commandMsg Message displayed while running Triage Client.


background, foreground Background and foreground colors of the applet.

progressBar Enables (1) and disables (0) the progress bar.

progressBarBackground, progressBarForeground Background and foreground colors of the progress bar.

Generating eSupport Packages

To generate an eSupport package:

- 1 Open an existing profile or create a new profile.
- 2 In the Profile toolbar, click **Package for WEB distribution** .
- 3 Select the folder where you want Triage to put the eSupport package and the HTML pages and other eSupport files.

Triage Console generates the eSupport package and copies all the eSupport files (such as the HTML pages) to a subfolder of the specified folder. The subfolder is given the same name as the profile.

Adding Files to the eSupport Package

If you need to include other files in the eSupport package (for example, a pre- or post-audit executable), edit the list of files in cab.txt. This file lists the files that will be put in the eSupport package along with the files that Triage Client automatically puts in the eSupport package:

- The application profile
- Any templates included in the profile

You can find the CAB.TXT file in the Triage installation directory.

If you want to copy any other files (for example, other HTML pages) to the folder where Triage puts all the eSupport files, edit the list of files in `esupport.txt`. You can find the `esupport.txt` file in the Triage installation directory.

Specifying the Triage Client Command Line

When you create an eSupport package, Triage generates an HTML page that contains an embedded Java applet that downloads and runs Triage Client. The Triage Client command line is specified in the HTML page as a parameter of the Java applet. For example:

```
<PARAM name=command value='TriageClient.exe /profile="NewProduct.trp" /mode="wizard" '>
```

The value of the **/profile** argument is the name of the profile. The rest of the command line is controlled by the eSupport preferences.

Web Arguments Value of this attribute is appended to the command line. Allows you to add any command-line argument.

Web Auto Email When set to True, adds `/automail="1"` to the command line.

Web Client GUI Mode Sets the `/mode` command-line argument, which specifies the Triage Client interface mode.

Web Email Address Sets the `/mail` command-line argument.

Web Show Mode When set to Minimized, adds `/hidden="1"` to the command line.

Web Smtip Server Sets the `/smtip` command-line argument

Chapter 8: Distributing Triage Client on CD

You can distribute Triage Client on the same CD (or disks) with your application files.

To build the CD distribution package:

- 1 In the Profile toolbar, click **Package for CD Distribution** .
- 2 Locate the folder where you want to put the distribution files. Triage copies all the required files to the specified folder.


The files required for CD distribution are:

- The application profile and any templates referenced by the profile.
- The files listed in cab.txt. You can edit this file to include other files in the CD distribution (cab.txt is located in the Triage installation folder).
- MqllsDiag.exe, if **Internet Information Services** is **True**.

Chapter 9: Configuring

Triage provides options for configuring Support Site, setting event logging options, registering products, and getting and revoking licences. These option settings are shared by all agents and consoles.

To set options:

On the Action menu, click Options, or click  on the console toolbar.

When you change any of the options, the changes apply to all agents and consoles. Agents are sent a notification message of the changes, and consoles pick up the changes the next time they start up.

Moving the Support Site

When you change the location of the Support Site, all consoles are automatically notified. If the notifications fail and the agents and consoles cannot automatically update their Support Site settings, they can do it manually through the Options dialog.

You can also move the data in your Support Site to another Support Site.

To move Support Site data to another Support Site:

- 1 In the Support Site Path box, enter the path to the other Support Site.
- 2 In the dialog that appears, click the Move all data from your current Support Site to the new location check box.

- 3 If you want agents and consoles installed from the current Support Site to switch to the other Support Site, click the Notify all clients of the change in Support Site location check box.

To switch to a different Support Site:

- 1 In the Support Site Path box, enter the path to the other Support Site.
- 2 In the dialog that appears, click the Notify all clients of the change in Support Site location check box.

When you don't select the Move all data from your current Support Site to the new location check box, you switch to using the other Support Site and its data.

The Support Site User Account

The Support Site user account is used to access the Support Site shared folder.

The Support Site user account must have the appropriate privileges on each local computer to perform tasks such as saving audit reports. The Support Site user should be a Domain Administrator that has local Administrative privileges on each computer.

To verify that the Support Site user account has access to the SupportSite from a computer, log on to Windows with that user account and try to copy a file to and from the Support Site shared folder.

Event Logging

By default, consoles log events on the local computer. On Windows NT, 2000, and XP, events are logged to the Event Log. On Windows 95, 98, and Me, events are logged to a text file.

On Windows NT, 2000, and XP, you can log all events to the Event Log on a central server.

To log all events to the Event Log on a central server:

- 1 Click the Log events to a central server check box.
- 2 Type the computer name of the central server.

By default, settings changes are applied only to new installations of Triage Console.

To apply new event logging settings to all installed consoles:

Before you click OK, click the Apply new settings to all Clients check box. Consoles pick up the changes at their next startup.

Licensing

Each Triage product (Triage, Triage Pro, and Triage/db) requires a serial number and a unique license key. The serial number is used to register a product. After you register a product, you can get license keys for the product. Typically, this is done during setup.

Registering Products

Sometimes you need to register a product after it has been installed. For example, you may installed a product but only later decided to purchase licenses.

To register a product:

- 1 Click the License Information tab.
- 2 In the Serial Number box, enter the serial number and click Register.

If you have an evaluation version of a product, you may need to unregister it when the evaluation period ends.

To unregister a product:

- 1 Click the License Information tab.
- 2 In the list of products, click the product.
- 3 Click Unregister.

To get a License Key for a product:

- 1 Click the License Information tab.
- 2 In the list of products, click the product.
- 3 Click Get License.

Revoking Licenses

If someone is on an extended vacation or leave of absence, you can revoke the license so someone else can use Triage.

To revoke a user's license for a product:

- 1 Click the License Usage tab.
- 2 Expand the product and click the user's computer.
- 3 Click Revoke.

Appendix A: Audit Preferences

Audit preferences are options and flags for controlling the behavior of Triage Client.

Audit Preferences

Attach Copied Files If **True**, all copied files binary or ASCII are included as attachments in the audit report. If **False**, ASCII files are embedded in the audit report and binary files are included as attachments.

Compress Audit Report If **True**, the audit report is compressed (.ZIP format).

File Version Information Determines how much version information Triage Client extracts from a file. Applies to all file types.

- Minimal extracts FileVersionProp, FileDescriptionProp, and LegalCopyrightProp.
- Normal extracts the Minimal information plus: CompanyName, InternalName, OriginalFileName, Productname, and ProductVersion.
- Full extracts Normal and Minimal information plus: Comments, FileVersion (not the same as the one above), ProductVersion (not the same as the one above), TradeMarks, PrivateBuild, SpecialBuild, fileFlagsMask, FileFlags, Os, Type, SubType, Translations, and Translations-Charset.

Use Minimal or Normal. Full extracts significantly more information for each file creating large audit reports. Large audit reports require more memory and affect performance.

Impersonate Password Password of the user to impersonate when running Triage Client.

Impersonate Name Name of user to impersonate when running Triage Client. If your application runs with the privileges of a specific user, you might want to impersonate that user to ensure Triage accesses the system data with the same privileges as your application.

Include CD-ROMs Search CD-ROM drives for files. Users can override this setting in Triage Client.

Include Network Drives Search network drives for files. Users can override this setting in Triage Client.

Internet Information Server If **True**, Triage Client collects information about Microsoft Internet Information Server (IIS).

Post-Audit Application, Pre-Audit Application

Names of executables to run before or after auditing the system. The names can include a Triage variable (for example, \$(InstallDir)\myutility.cmd). Pre- and Post-audit applications must be windowless applications.

If just a file name is specified, the executable must be located in the user's path or in the current directory.

To specify more than one application, use semicolons (;) to separate the application names.

Specify command-line arguments immediately after the executable name (for example, myutility.cmd /argument).

Recursive Registry Scan If **True**, Triage collects all subkeys of the registry keys specified in the profile. If **False**, Triage collects just the specified registry keys.

WMI Core Package URL URL for downloading WMI core components.

Client Preferences

Auto Exit (Classic mode only)

If **True**, Triage Client will automatically exit when completed. This attribute is ignored if the profile includes self-repair rules.

Auto Save Name and path of the audit report to be automatically saved when the audit is completed. Variables can be used to specify the path.

Example: c:\TriageAudit.cab

Example: \$(TriageDir)\TriageAudit.cab

Auto Start Audit (Classic mode only)

If **True**, the audit is automatically started once a profile is loaded. The user does not need to click on the Start button. This flag is ignored when Triage Client is in Wizard mode.

Desktop Shortcut Description When Triage Client exits, it can create a shortcut to a URL (as specified by the Desktop Shortcut Url attribute) on the user's desktop. The Desktop Shortcut Description attribute sets the description of the shortcut as it will appear on the user's desktop.

Desktop Shortcut Url When Triage Client exits, it can create a shortcut to a URL. The Desktop Shortcut Url attributes sets the http address of the URL.

Enable Audit-Delete If **True**, users can delete information from audit reports. The audit report viewer in Triage Client has a Delete button that is displayed only when this flag is set to **True**.

Enable Save Report (Wizard mode only)

If **True**, users can access the Save button and save the audit report to a file of their choice.

Finished Message (Wizard mode only)

Customizable message displayed in the Finished dialog of Triage Client after the user sends the audit report.

Help File URL The URL for the location of the TriageClient.htm help file that is specific to this profile. This help file is used when the user clicks on the Help button in Triage Client.

Request User Info (Classic mode only)

If **True**, Triage Client prompts the user to enter their personal information and a problem description before it saves or sends an audit report.

Show Audit Results If **True**, the Audit Results are automatically displayed at the end of the audit. If **False**, the user needs to click on the View Edit Report button to display the Audit Results.

In Wizard mode, Triage Client automatically displays the View Audit Report dialog when the user clicks Next on the Problem Description dialog (or, if the Problem Description is not visible, when the user clicks Next on the User Information dialog).

eSupport Preferences

These settings are used to build the Triage Client command line when you package Triage Client for eSupport.

Web Arguments Specifies additional arguments to add to the Triage Client command line.

Web Auto E-mail If **True**, after Triage Client is downloaded and automatically run, the audit report is automatically e-mailed. Set Web Auto E-mail to **False** if you want users to view and edit the audit report before e-mailing it.

Web Client GUI Mode Sets the interface mode of Triage Client. Triage Client has two basic modes: Wizard and Classic. Wizard mode guides the user through a series of steps, while Classic mode provides a single main dialog that allows the user to perform actions in any order. Users can switch between the two modes when Triage Client first starts, unless you use the ClassicOnly or WizardOnly modes.

Web E-mail Address After Triage Client is downloaded and automatically run, this is where it sends the audit report.

To specify more than one e-mail address, use a semi-colon (;) between each e-mail address.

To have Triage Client prompt for the e-mail address at runtime, specify a User Defined Variable as the e-mail address.

Web Show Mode Set Web Show Mode to Minimized if you want the Triage Client window minimized when Triage Client is automatically run. The Maximized setting is not supported.

Web Smtip Server SMTP server used when sending the audit report after a download.

Mail Preferences

Enable Mail Options (Wizard mode only) If **True**, users can access the Mail Options property dialog. You can typically disable the Mail Options if you have specified an SMTP server in

your profile attributes. In that case, you need to ensure that your SMTP server is setup to allow anonymous users to send mail via that server.

Send Error Message (Wizard mode only) Customizable message displayed if the Send operation fails. An option to save the audit report is always appended to the message.

Smtip Server SMTP server used by Triage Client when e-mailing the audit report. If you leave this field blank, Triage Client will automatically detect the SMTP server by scanning the registry.

Security Preferences

Include File Security If **True**, security information for all the selected files is collected by Triage Client.

Include Registry Security If **True**, security information for all the selected registry entries is collected by Triage Client.

Include Share Security If **True**, security information for all the shares found in the Network Neighborhood is collected by Triage Client.

Self Repair Preferences

Action column width Width (in pixels) of the column that holds the Fix it button in the Self-Repair Diagnosis dialog of Triage Client.

Diagnosis Show Details If **True**, Triage displays problem details.

Enable Self-Repair If **True**, Triage Client uses the self-repair information in the profile to diagnose and fix problems. Otherwise, Triage Client ignores all self-repair information in the profile.

Note that self-repair can be disabled for individual objects (such as files, shortcuts, and ActiveX controls).

Executable Name The executable that Triage Client tries to run when the user tests the fixes made by Self Repair.

Run Self-Repair (Classic mode only)
If **True**, Triage Client automatically pops up the Self Repair dialog after the audit if any problems were found. Otherwise, the user must click the **Diagnose and fix** button to run Self Repair.

Update Profile If **True**, Triage Client updates the self-repair rules in the profile and creates a self-repair package.

If **False**, the /updateprofile command-line switch determines if Triage Client updates the self-repair rules in the profile and creates a self-repair package

Phone Number Phone number the user can call for information or support. In Triage Client, this information appears only in the audit report viewer.

Product Name Name of the product. Used as the default name for the profile.

Product Version Name and version of the product for which the audit report was generated. In Triage Client, this (read-only) information is the default information displayed on the User Information dialog.

Product Preferences

This information is included in the header of the audit report. Some of this information is also available to the end user from the Options dialog of Triage Client.

Company Name Name of the vendor.

E-mail Address Default e-mail addresses used by Triage Client to send audit reports. In Triage Client, users can change these e-mail addresses on the E-Mail tab of the Options dialog.

Use a semi-colon (;) to specify more than one e-mail address.

To have Triage Client prompt for the e-mail address at runtime, specify a User Defined Variable as the e-mail address.