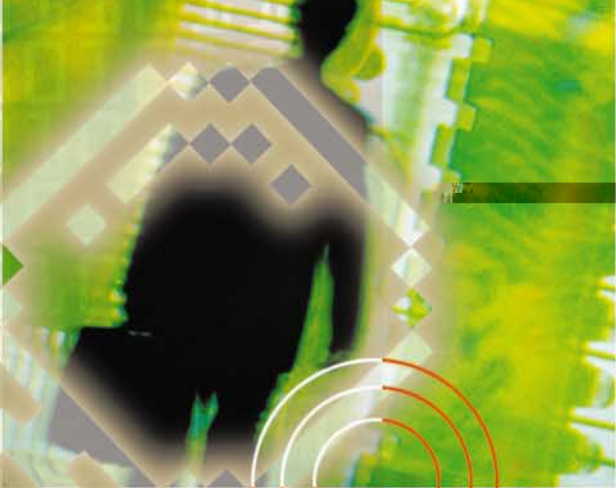




Trace Analyzer™ for Java™ by Golden Code Development is a graphical user interface which allows you to decode and interpret trace file data. The flexible and powerful companion product to Network Trace™ and Kernel Trace™, the Trace Analyzer also reads trace files generated by other popular tracing software, preserving your previous investments in tracing tools.

# Trace Analyzer™

**for  
Java™**



**Configurable reporting and filtering let you control the view of your data**

**Flexible architecture supports both network and non-network traces**

**Platform independence offers the widest range of deployment choices**

## Trace Analyzer for Java

is designed to be extensible and customizable. Written in Java to be platform neutral, the Trace Analyzer offers a rich set of features in a simple interface. New report formats and data filters are easily added by the end user to provide a view into your traces never before available.

**Power.** Comprehensive protocol decodes and a host of useful, preconfigured reports and data filters make the Trace Analyzer a powerful analysis tool, even for the first-time user. For the more advanced user, the detailed structure of every protocol is

exposed, enabling the creation of extremely robust reports and data filters. Symbolic name aliasing substitutes names of the user's choice for numeric addresses, to enable more meaningful and readable reports.

**Flexibility.** With a customizable report and data filtering system, the Trace Analyzer allows a user to view trace data in completely new ways. No longer are analysts limited to a small set of immutable, preconfigured reports. If one of the standard reports does not meet an analyst's needs, it is easily enhanced or replaced. Robust data filters are easily created using a simple expression syntax and optional user functions, in order to distill specific information from a trace. The same filters are used for search functions, reducing the overall effort needed to use the system.

**Versatility.** Written in 100% Java, the Trace Analyzer is designed to run on multiple platforms "out of the box", giving you deployment flexibility. Because it takes advantage of the cross-platform user interface library provided with Java 2, the user experience is consistent across platforms, reducing training time and cost.

The screenshot shows the Golden Code Trace Analyzer v1.0.0 interface. The main window displays a table of trace data with columns for Mark, Frame, Size, Delta Time, Source Address, Destination Address, and Summary. The table contains 31 rows of data. Below the table is a 'FILTER SELECTOR' section with a dropdown menu set to 'No Filter'. To the right of the table is a 'DETAILED REPORT' section showing a hex dump and ASCII representation of the selected frame. Below the hex dump is a 'HEX DUMP REPORT' section showing the raw hex data. The interface also includes a 'Show hidden columns' checkbox and a 'SELECT' dropdown menu.

## Support and Service

[www.goldencode.com](http://www.goldencode.com)

**E-mail support for Trace Analyzer is provided for one year from the date of purchase at no additional charge. Targeted response time is within one business day. The Golden Code website provides the latest product tips, updates, fixes, and documentation.**

**Phone support contracts are available for an additional fee.**

**On-site consulting services are available at an additional charge to assist with planning, installation, deployment, problem determination, and skills transfer.**

**Ease of Use.** The features of the Trace Analyzer, while powerful, were designed also to be accessible and easy to use. Reporting, filtering, and searching options are available directly from the main interface. New data filters can be generated automatically with a double-click of the mouse. Context-sensitive help is integrated into the application.

## Features

- **Support for Network and Non-Network Protocols.** Designed to be a general purpose analysis tool, the Trace Analyzer is not limited to network analysis. For example, in addition to broad network protocol support, v1.0 offers optional protocol support for Kernel Trace for OS/2, a Golden Code API tracing product for the OS/2 platform.
- **Customizable Reporting System.** Overview Reports are completely configurable, giving the user unprecedented control over the presentation and format of trace information. Existing reports can be changed or removed. New reports can be created to represent any combination of data from multiple protocols.
- **Flexible Data Filtering.** Using a straightforward and flexible expression syntax, the user can subset the data within a trace to include only those records which meet a defined set of criteria. Data filters can be created using an interactive user interface or can be automatically generated with a double click of the mouse.
- **Flexible Search Engine.** The same set of tools used for data filtering is used to provide a powerful search capability, requiring less of a learning curve.
- **Printing with Preview.** Reports can be printed in color or black and white. A useful print preview feature is provided to customize your print job and eliminate wasted paper.
- **Symbolic Name Aliasing.** Numerical addresses can be replaced with symbolic names meaningful to the user.
- **Unlimited Bookmarks.** Any number of individual records in a trace file may be bookmarked. Bookmarks can be used to create special report views and to define the contents of print jobs.
- **File Save and Export.** Any subset of trace file records can be saved in the format of the original file. Overview Report data can be exported for further manipulation as comma- or tab-delimited text.
- **Export and Import Tools.** Overview Report templates and data filters can be exported and imported for sharing or archival purposes.
- **Summary Information.** Useful summary information about a trace file is presented in its own report.
- **Configurable User Interface and Data Representation.** The user interface itself can be customized by the user, including the formats used to represent certain data types in reports.
- **Online Help.** Searchable, context sensitive, online help is just a keystroke away. All help content is available in a separate, softcopy format for review outside of the Trace Analyzer program.

## Choosing the Right Edition

The Trace Analyzer is available in three editions, depending upon your needs:

### STANDARD EDITION

This edition of the product has all protocol support enabled. This includes network protocols and Kernel Trace for OS/2 protocols. This edition is intended for those who wish to read trace files generated by Network Trace for OS/2 or other network packet capture products, and optionally, trace files generated by Kernel Trace for OS/2.

### KERNEL TRACE EDITION

This edition of the product supports only the Kernel Trace for OS/2 set of protocols. Thus, this edition can only read trace files generated by Kernel Trace, and will not read network oriented trace files. All other product functions are enabled.

### EVALUATION EDITION

This edition of the product is capable of reading all of the protocols supported by the Standard Edition. It is available at no charge for the purpose of evaluating the product before purchasing a license. Because it is free, this edition will time out after a certain period, and may have functional limitations as well.

# Technical Specifications

## Supported Platforms

The Golden Code Trace Analyzer is supported on the Java 2 platform, on the following, tested Java Virtual Machines (JVMs):

- IBM v1.3 for OS/2
- Sun Microsystems v1.3 for Windows 95, 98, NT, 2000

Although this product may function properly on other JVMs, support for such platforms will be provided on a best efforts basis only. Golden Code will endeavor to expand the list of tested JVMs over time.

*Note: This product will not function properly on versions of the JVM previous to version 1.3.*

## File Formats and Protocol Support

Please refer to the online documentation for a list of the protocols and file formats currently supported by the Trace Analyzer. Support for some protocols is partial, and this support is constantly evolving. Please be sure to take advantage of our free evaluation period to ensure that the Trace Analyzer meets your needs.

## Hardware Requirements

The following are the minimum and recommended hardware requirements for the Trace Analyzer target workstation:

	Minimum	Recommended
CPU	200 MHz Pentium	400+ MHz Pentium II
RAM	96 MB	128+ MB
Disk Space	10 MB	10 MB
Screen Resolution	1024x768	1024x768 or higher
Colors	256	65535+

**CPU.** Certain operations within the Trace Analyzer are CPU bound (e.g., data filtering and find operations). Additionally, the Java user interface libraries used presume fairly modern hardware, and may seem sluggish on slower processors. The impact of a slower processor will be particularly evident when pr

## Statement of Year 2000 Compliance

When properly used in accordance with its associated documentation, the Golden Code Trace Analyzer will correctly store, display, process, provide and/or receive date data from, into and between 1999 and 2000 and the twentieth and twenty-first centuries, including leap year calculations, provided that all other technology used in combination with the Golden Code Trace Analyzer properly exchange accurate date data with it.

© 2000, 2001 Golden Code Development Corporation. All Rights Reserved.

Golden Code, the GC logo, Kernel Trace, Network Trace, Trace Analyzer, and Practicing the Science of TCO Reduction are trademarks of Golden Code Development Corporation.

IBM and OS/2 are registered trademarks of International Business Machines Corporation. Java and Java 2 are trademarks of Sun Microsystems, Inc. Pentium is a registered trademark of Intel Corporation. Any other named products or brands referenced herein are the property of their respective owners.

## Golden Code Development Corporation

1455 Old Alabama Road, Suite 135  
Roswell, Georgia 30076

TEL: 678-352-2301 FAX: 678-352-2305

E-MAIL: [info@goldencode.com](mailto:info@goldencode.com)

[www.goldencode.com](http://www.goldencode.com)