BRISTOL

T·E·C·H·N·O·L·O·G·Y

eSleuth™

eBusiness Transaction Analysis Software
that Simplifies the Development of
Reliable, High-Quality eBusiness Systems

A Technical Description for Software Developers and Development Managers

Build eBusiness Systems with Confidence

You've invested heavily in high-availability equipment and middleware to provide reliable software communications. But one thing worse than your eBusiness system going down is your eBusiness system staying up and silently mishandling your business transactions.

eSleuth™ is transaction analysis software that ensures your eBusiness information gets to the right place, at the right time, with the right content. eSleuth detects and helps alleviate information flow failures in MQSeries-based eBusiness systems running across UNIX, Windows NT, Linux, and OS/390. Using eSleuth during distributed application development shortens the time to deployment of new eBusiness systems and improves their reliability, performance, and quality by ensuring that eBusiness transactions are completed successfully and efficiently.

eSleuth Maximizes Your Middleware Investment

With the phenomenal growth in eBusiness, middleware such as MQSeries has become the linchpin of the corporate computing environment. Yet, despite the pivotal role of middleware, solutions for analyzing inter-component events for the purpose of identifying and resolving logic flow problems and performance bottlenecks are sorely lacking. eSleuth fills this void by graphically analyzing eBusiness system transactions across system and application boundaries, enabling you to visually pinpoint information flow failures and performance bottlenecks. Only eSleuth lets you look inside your application to identify the exact logical cause of information flow failures.

The complexity of distributed eBusiness systems makes it virtually impossible to determine the cause of a failure. eSleuth provides a unified view of your complete eBusiness environment. This high-level view enables you to clearly see the logical flow of information throughout your entire system, making it easy to see information flow failures and performance bottlenecks.

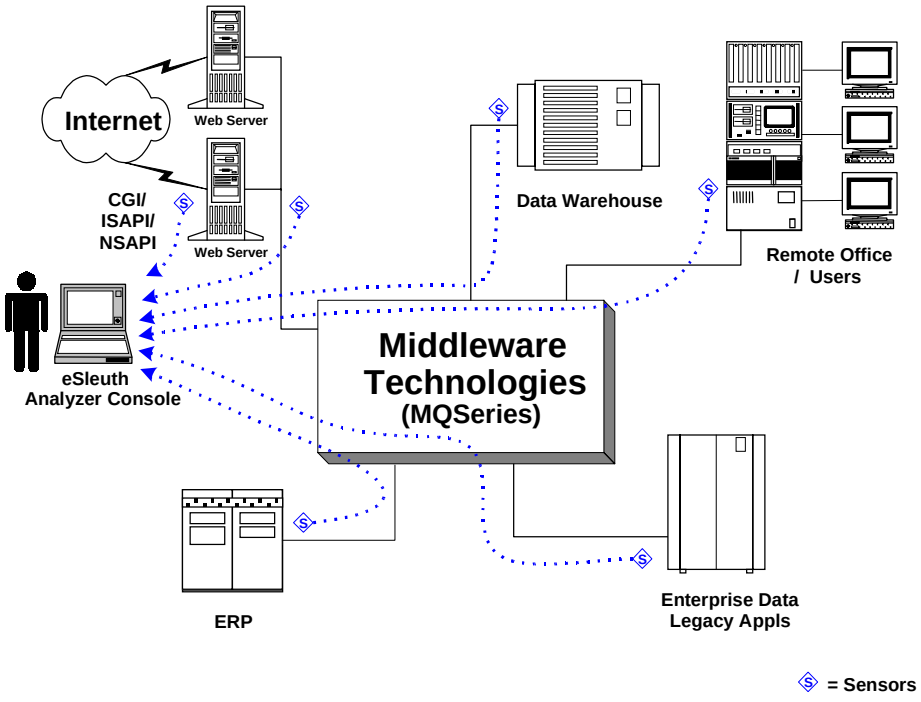

Figure 1: eSleuth offers a single point of problem isolation across eBusiness environments by analyzing and correlating transactions for visual presentation.

The initial version of eSleuth supports organizations using IBM MQSeries message-oriented middleware. Future versions will analyze additional technologies such as HTTP, CGI, CORBA, COM/DCOM, and Microsoft Message Queue (MSMQ).

More Than Monitoring: Complete Transaction Analysis

eSleuth enables you to identify problems during development and provides the information you need to resolve these problems—before your system goes into production. eSleuth traces MQSeries API calls across distributed processes and graphically displays the flow of messages between processes and queues. By analyzing the details of each API call, eSleuth enables you to drill down into application-specific data, down to data structure contents.

While middleware monitoring tools alert you to MQSeries communication failures, eSleuth has some key features that distinguish it from existing middleware management and monitoring tools:

- eSleuth provides *logical* diagnostic information to developers (such as API calls, call arguments, return values, etc.), while management and monitoring tools focus on *system* data such as queue status.

- eSleuth correlates API calls made from different components in the system to form a complete *transaction* view; including graphical depiction of the entire system.

eSleuth generates and stores information for each MQSeries API (MQI) call made by an eBusiness system. Even though MQI calls may occur asynchronously, on different hosts, and within different running programs, eSleuth correlates the calls that relate to the same business transaction in your application and graphically presents that information. For example, eSleuth connects the MQGET call that retrieves a message with the MQPUT call that sent it, enabling you to trace the logical flow within your application across program, execution thread, queue manager, or host boundaries.

In addition, eSleuth translates MQI details into understandable values that you can interpret and act on immediately. It formats data structures (MQMD, MQPMO, etc.) and present the symbolic flag names (MQGMO_SYNCPOINT, MQOO_INPUT_SHARED, etc.) for MQI data values, even if your program is compiled with optimization. You can see the user data buffer within each message as well. Moreover, eSleuth allows you to control which MQI calls are reported to the eSleuth Analyzer by specifying filtering criteria. This filtering capability enables you to focus on problem areas.

While message monitoring tools can analyze the messages currently available on queues, eSleuth captures and logs all the messages. This enables you to see events that may have persisted on the queue for a very short period of time and would have been missed by traditional monitoring tools. Seeing how a transaction fails is a requirement for efficient testing and debugging of transactions. A message that appears to be correct may actually be the cause of a failed transaction; but the ability to link transaction events to finding this problem can be time consuming. eSleuth's automatic generation of the entire transaction help you quickly identify these types of problems.

Only Analyze the Data You Need to Solve a Specific Problem

Through the use of eSleuth presentation filters, you can view the collected information in a variety of ways, and from each view you can drill down into more event details such as the MQ message descriptor and user data. Instead of simply seeing streams of numbers and return codes, eSleuth translates the information to you into understandable values. You can decide to view all the collected events as transactions. eSleuth automatically correlates the collected events into business transactions. In addition to the transaction view, you can view the events as they occurred sequentially from all the applications and hosts being monitored. You can narrow your search by time, MQI call, queue, queue manager, host, process, thread, and a number of other criteria. For complex, organization-specific criteria, you can even program your own custom filters.

Solve the Challenges of Distributed Transaction Analysis

The following table demonstrates how eSleuth addresses problems in testing and debugging a distributed transaction system.

The Problem	How eSleuth Addresses the Problem
Development and diagnostic skills are required on many platforms.	eSleuth provides a central transaction analyzer with a Microsoft Windows user interface. Working from a central console, you can view transactions being processed on local or remote heterogeneous computer systems.

Visualizing transaction flows through a complex eBusiness system is difficult.	eSleuth provides a component layout graphical view, providing a view of transactions occurring across different systems. eSleuth automatically determines the transaction flow by monitoring a running system; no developer interaction or code changes are required.
Tracing transactions into legacy or third party applications is impossible.	By providing views of the MQSeries API calls and transactions resulting from these calls, eSleuth provides a glimpse inside the black boxes and sees what MQSeries API calls are being made (including the ability to see the API parameters). This is done without access or modification to the source code of these legacy applications and components.
Asynchronous systems process thousands of transactions per second	eSleuth's ability to filter the information gathered and presented from the remote systems allows you to isolate only the data needed to identify problems that are occurring within transactions.
Middleware technologies hide diagnostic information.	Presentation filters within eSleuth enable you to view all of the middleware-specific data fields in addition to other information that is normally transparent to the developer.
Middleware monitoring does not provide enough information.	eSleuth provides a view showing all the MQSeries API calls made in the distributed system, in addition to the correlation of these API calls into distributed transactions.
The cause of performance bottlenecks is hard to pinpoint.	With eSleuth, transaction path tracing is done at the API level within software components of the system. Event sequences can be analyzed at the thread level within these components. eSleuth maintains timestamp information on all gathered events, including corrections for clock skew between the nodes in a distributed environment.

How eSleuth Works

eSleuth consists of two major components:

- Platform- and technology-specific **sensors**, which capture information about each API call
- An **analyzer console**, which stores captured transaction data, analyzes transactions, configures filters, and presents the results graphically

eSleuth Sensors

eSleuth sensors are light-weight, non-intrusive monitors that trace API usage across middleware technology such as MQSeries. Available for all major platforms, eSleuth sensors are installed on each host in the distributed application environment to capture transaction data for each MQ API executed on that host.

The type and amount of information captured by each eSleuth sensor is easily programmed from the eSleuth Analyzer console. Data collection filters, configured through easy-to-use dialogs, control the data each eSleuth sensor collects. For example, you may only want the MQMD information on some hosts, MQMD information and user data on other hosts, or all details from all the hosts.

Unlike many other debugging or application troubleshooting methods, you do not need access to the original application source code to install and use eSleuth sensors. The eSleuth sensors monitor your system unobtrusively. Once installed on the host, eSleuth sensors are able to collect information on every MQI call that is made. Even if you do not have source code to certain applications, eSleuth sensors can still trace all the API calls.

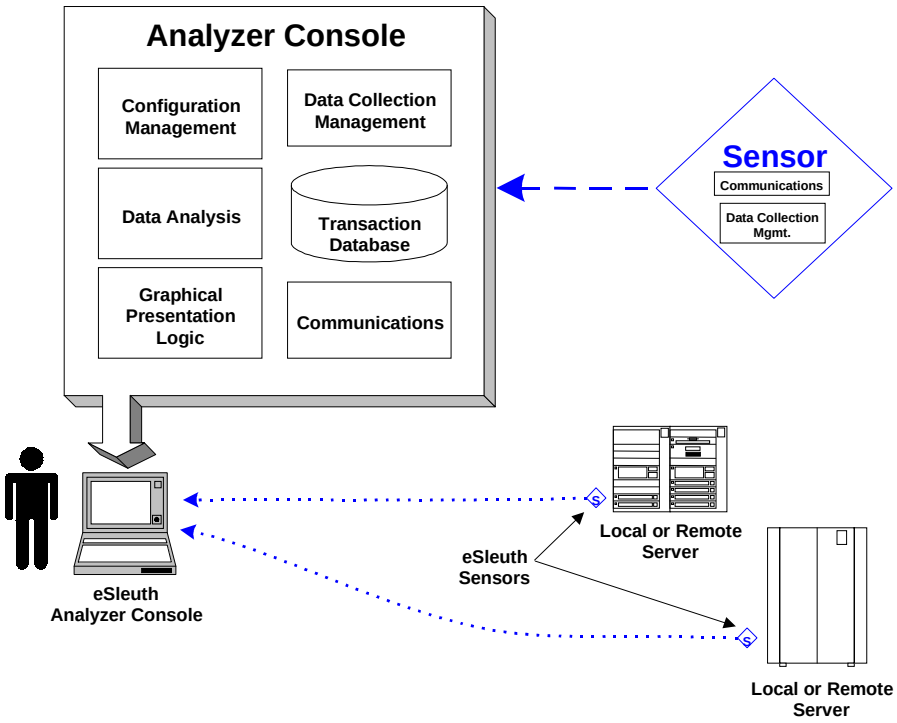

Figure 2: eSleuth sensors relay distributed transaction data to the Analyzer console for analysis, correlation, and visual presentation.

eSleuth Analyzer Console

The eSleuth analyzer console is the single point of interface for diagnosing problems in your distributed MQSeries application. The analyzer console receives messages from the installed eSleuth sensors, processes them, and displays different views of this data to help you diagnose problems within your distributed system. An embedded database is included with the console for easy setup and efficient storage of data received from eSleuth sensors.

The eSleuth analyzer console provides four primary views: the Component Layout, Dynamic Transaction Visualization, Event History, and Event Details.

Component Layout

The component layout view (Figure 3) graphically displays the components of your distributed system, including the message queues, hosts and processes involved, and which process is talking with which queue. Arcs link the queues to the processes. The thickness of these arcs indicates performance characteristics of your system.

Dynamic Transaction Visualization

The dynamic transaction view (Figure 4) displays the transactions as they happen across multiple hosts and operating systems. You can apply presentation filters to display only the events and transactions applicable to any particular transaction for rapid analysis of transaction problems. No user interaction is required to generate these transactions—eSleuth automatically links together the messages that make up the same business transaction and displays them graphically.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.