

The White Papers

Achieving Application Availability, Performance and Service Levels Using Foglight[®] 3

By Dennis Hejna

Audience and Purpose

This paper presents the challenges and solution for monitoring availability, performance and service levels of business applications in ERP, CRM and e-business environments. It is intended for system and application executives, managers and administrators responsible for system and application availability, performance, capacity planning and troubleshooting. It is relevant in the following IT technical infrastructures: corporate MIS, application and web hosting and management, and general e-business.

Contents

<i>Application Availability and Business Challenges</i>	3
<i>Need for Application Monitoring</i>	5
<i>IT Organizations and Industry Monitoring Methodologies</i>	5
<i>Approaches to Measuring Availability, Performance and Service Levels</i>	8
<i>The Foglight Approach to Availability, Performance and Service Levels</i>	10
<i>Foglight Architecture, Features and Differentiators</i>	15
<i>Summary</i>	18
<i>About the Author</i>	19
<i>About Quest Software</i>	19

Achieving Application Availability, Performance and Service Levels Using Foglight® 3

By Dennis Hejna

Application Availability and Business Challenges

One of the major demands placed on IT professionals today is to keep vital applications available to users around the clock, and around the world, while holding the line on spending. Especially challenging is *proactively* ensuring application availability and performance, as users increasingly expect the IT organization to correct problems before they reach critical states. Complicating these demands is the complex and rapidly evolving architectures of ERP, CRM and e-business applications. Comprised of an interdependent set of networks, firewalls, servers, operating systems, processes, logs and application-specific managers, a critical or fatal event impacting any of these numerous interdependencies can cause an application outage or performance degradation.

Consider a situation where an IT technical administration group is responsible for ensuring several production and development applications over a large infrastructure consisting of dozens or hundreds of servers. If users start to experience availability or performance problems, administrators would have to investigate, manually or with automated tools, the configuration, availability and performance of the multiple technology layers and components, specifically:

- Availability and utilization of network servers and services
- Availability and resource consumption of critical processes and resources
- Ability of other users to connect to and/or run the impacted transactions
- The status and interpretation of system and application log files
- Web and application server configuration, activity and status
- Database, tablespace and SQL performance

IT administrators need the ability to go back in time to conduct a proper investigation of the application stack at the time a problem develops. Investigating real-time information is often too late and may not reveal useful information.

Another issue is the time it takes for an administrator to sift through copious amounts of information to identify areas on which to focus. With numerous applications and technologies deployed, it is unreasonable to expect administrators to be expert in more than a few and hence be able to readily identify potential problems.

It is the responsibility of the IT organization to implement the methodologies and measurements that show whether an application is available and performing to agreed-upon levels. A summary of the issues that should be considered when developing an application management methodology include:

The Application Technology Stack

The application technology stack is the sum of the layers, servers and other things that make up an application, including Web, application and database servers, underlying processes, object managers, application logs and end-user performance. All of these things should be correlated into a logical picture to determine an application's underlying layers and components to show application availability, at a glance, and to identify applications impacted by general system events.

In addition, there may be more than one application represented in an application's technology stack. For example, a PeopleSoft application that integrates with Siebel for pricing information, or with a custom application to generate purchase recommendations, would represent an extension to that technology stack.

End-User Transaction Availability and Performance

The technology stack includes a layer to track end-user performance. Can users connect to and complete a transaction from various geographic locations? Even if the underlying parts of an application appear normal, the bottom line is whether users can complete transactions. If a user can complete a transaction, it logically follows that all underlying parts needed by that transaction are functioning normal.

Application Knowledge

ERP, CRM and other e-business applications integrate various proprietary and commercial technologies, including Web, application server and database products. Expertise is required to determine how each of these technologies work and to assess operational issues affecting availability and performance.

24x7 Availability

Globalization, consolidation, and user demands for extended hours of operations are forcing IT organizations into round-the-clock operations for applications. IT organizations are required to develop procedures to manage these increasing responsibilities and availability windows without adding substantial staff.

Root-Cause Analysis

Access to both detailed real-time and historical information on any technology layer can be difficult if not impossible to obtain. The ability to determine the state of an environment at the time of a critical or fatal event; going back minutes, hours, days or weeks old, expedites problem analysis by placing focus where it belongs.

Application Service Levels and Return On Investment

Increasing corporate investment in vital ERP, CRM and e-business applications have led to the adoption of Service Level Agreements (SLAs), employed by IT organizations and user management as a tool to define, measure, and report application availability and performance. SLAs can be labor-intensive to produce, as the underlying data must be captured and correlated and the report itself generated. While SLAs are part of measuring the effectiveness of an application,

IT professionals and users may be reluctant to implement due to the effort required to maintain them.

Need for Application Monitoring

Application management methodologies employed by organizations differ markedly depending on corporate culture, business mission and size. Some organizations rely almost exclusively on custom-developed scripts that use operating system utilities such as ping, vmstat, df, top, perfmon, and so on, while others have implemented methodologies of varying levels of sophistication.

Custom-developed solutions may be sufficient and cost-effective in single-site environments with low transaction volume where the impact of downtime or performance degradation is limited. But, in larger organizations, the cost of developing, testing, documenting and maintaining those scripts, and developing various reports, may be surprisingly costly.

If the impact of downtime and performance degradation is significant and user satisfaction levels must be met, the need for an advanced 24 x 7 application monitoring solution that allows administrators to *proactively* monitor the entire application technology stack with many out-of-the-box features becomes very compelling.

This white paper discusses the application monitoring market and how Foglight® can help deliver high availability and performance to your users *efficiently* and *cost-effectively*.

IT Organizations and Industry Monitoring Methodologies

IT organizations responsible for system and application technical infrastructures can be categorized into the following organizational types:

Organization Type	Characteristics	Key Requirements
Traditional IT Infrastructures	Traditional IT management operations vary greatly in size and may include a network operations center and/or system staff. These organizations are tasked with monitoring a wide range of servers, applications and interfaces and consult heavily with upper management and end users.	24x7 or near 24x7 regional or global monitoring, encryption, breadth of technology coverage, ease of installation and deployment.
E-business/Internet Infrastructures	A dot com-type company, or a separate e-business unit of an established corporation, manages servers and applications accessed	24x7 remote monitoring, Web-based monitoring and usage statistics, firewall issues, encryption, rapidly changing

	by Internet users.	technology, ease of installation and deployment.
Management Server Providers (MSP)	The MSP manages servers and applications located at remote customer sites, although some servers may be local. Typically employs a network operations center (NOC) at a central location to monitor servers and applications and alert on critical and fatal events. Automated management tools are deployed at each site.	24x7 remote monitoring, ease of deployment to multiple sites, maintenance, encryption, security, service level reporting, and breadth of technology coverage.
Application Server Providers (ASP)	The ASP manages servers and applications located in a national or regional operations center. Typically employs a network operations center (NOC) to monitor servers and applications and alert on critical and fatal events. Automated management tools are normally deployed.	24x7 remote monitoring, ease of installation and deployment, service level reporting, depth of application coverage, ability to manage production and development applications with numerous instances.
Consulting	Consultants are engaged to perform functionality, integration, load testing and benchmark testing in production or production-simulated environments.	Ease of installation, configuration and deployment, trend and capacity reporting, depth of application coverage.

While all the above organizations have different business missions, they have many needs in common, among them:

- Immediate notification of critical or fatal events on any part of the application technology stack via e-mail and pages, and an option to automatically start corrective actions
- Real-time and historical capture of raw performance data for troubleshooting and reporting
- Quick installation, configuration and deployment on monitored servers
- Breadth of technology and application coverage due to heterogeneous flavors and releases of operating systems, Web servers, database servers and applications

Monitoring Methodologies

IT organizations use a combination of methodologies and products to monitor and diagnose application technical infrastructures. These include framework and network monitoring products, application monitors, application vendor-provided tools and command line utilities, sniffers, Web and database analysis tools, and custom scripts.

To better understand the purpose of each of these tools and how some organizations have deployed them, it is useful to categorize them as Level I, II or III, as discussed below.

Level 1: Network or Framework Monitoring

Network monitoring solutions have been used for many years and are designed to monitor very large, heterogeneous networks 24x7 to provide event management, the up/down status of servers and network connections, system logs, network interfaces, and basic system performance. These solutions are used to determine system availability or to alert if performance thresholds have been exceeded. Level 1 is sometimes associated with a Network Operations Center (NOC). Note that application monitoring does not occur at this level.

If a node or network connection fails, the NOC may not know which applications are impacted. There is neither any grouping of the interdependent components of an application nor in-depth monitoring of an application's detailed components such as object managers, application logs, SQL activity, and so on.

Level 2: Application Monitoring

Application monitoring solutions vary widely in their coverage and depth. Some application monitors are designed to monitor all the interdependent parts that make up an application stack to provide event management (alerts, etc.) and store real-time and historical information. Others are designed to monitor specific parts of the application stack, frequently the database and network layers.

While there is some overlap between Level 1 and 2, particularly with network and basic operating system monitoring, Level 2's purpose is to show availability by application (by grouping all the pieces together), depth of application collection and historical application performance. If a server or network connection fails, or if a critical error appears in an application log or if an object manager fails, the administrator will know which applications are affected.

Another difference between Level 1 and 2 is historical data collection. Level 1's main purpose is real-time network and system event management, so it will typically keep a history of *events* only (There are some exceptions in that add-on products may be available to maintain some historical data collection). Application monitors (such as Foglight) maintain availability, performance and capacity information in a central repository for days, weeks, months or other user-defined

period, whether or not it triggers an alert. This data is used for a variety of purposes, including trending, SLA and capacity reporting, and troubleshooting.

Foglight is a Level 2 solution. It is designed to provide application availability and performance monitoring and historical data collection. It can be deployed as a Level 1 tool in some cases, or as a complementary solution to a Level 1 framework, as Foglight provides out-of-the-box and customization integration facilities to send events to Level 1 solutions.

Level 3: Diagnostic Tools

Diagnostic solutions are launched as needed to diagnose performance problems of a specific layer or software component in real-time. These solutions take frequent data collection samples of a broad array of metrics over a limited period of time. Level 3 products are not intended to operate over long-term intervals. Network sniffers, application tracing products, debugging tools, and database diagnostic products are categorized as Level 3.

Quest has a broad range of Level 3 diagnostic tools including the popular Spotlight™ diagnostic product line and other network, Web and database tools.

Approaches to Measuring Availability, Performance and Service Levels

When it comes to application availability, users know exactly what they want: continuous access to any transaction from any location with predictable performance. Establishing monitoring methodologies and service level measurements is an art, not a science, and requires considerable focus from IT and user management.

The key question that IT management and users should ask is:

“How should application availability be defined in our environment?”

To answer that question, one must consider the application technology stack. All applications can be broken down into component layers as shown in the diagram below. The layers are ordered, from top to bottom, from specific to general. The bottom layers can be thought of as the foundation, the middle layers as the structure, and the top layers as the “acid” test.

Layer	Description
Business Transactions	The “acid” test that shows whether the transaction works. A transaction is repeatedly played back and measured to determine availability and response time, from various geographic locations, inside or outside the firewall. Foglight’s RTM (Response Time Measurement), SiebelBot and PeopleBot are designed to provide business transaction measurement.
Application Integrations	3 rd party component integrations such as ERP to ERP, ERP to CRM, ERP to data warehouse, Web solutions and other 3 rd party applications.
Applications	Object managers, application log files, performance flags and metrics, and SQL activity associated with commercial and user-developed software packages (e.g. Web, application and database servers, ERP, CRM, e-business).
Operating Systems	Servers, operating systems, and resources (CPU, memory, disk) that the processes running on them depend upon. Also includes system log monitoring to capture critical and fatal errors such as a bad disk, corrupted memory blocks, SCSI bus failure, 3 rd party storage manager error.
Network And Services	Server availability and communication pathways (local and wide area) for all the above layers. This layer includes “Networked Services” which are services that are offered over the network such as LDAP, NFS, SMTP, PING, NSLOOKUP, database connections, and other port/listeners with request/response interaction. If the network or service is down or degraded, application clients service levels cannot be met.

Illustration 1. Foglight’s approach to application availability using the application stack.

Indirect and Direct Availability Measurements

Observation of the availability of the interdependent components of the first four layers – network, operating system, applications, application integrations – implies that the application should be functioning normally. Yet, these observations are indirect ways of measuring service level performance. System components can look perfectly healthy when an application becomes unavailable. Only the measurement of a user’s experience can provide the complete availability picture.

Indirect measurement is useful in determining performance and trends and in monitoring CPU, memory and other resources that affect current and future performance and planning capital expenditures. It also allows administrators to go back in time to determine the state of the environment when a business transaction has degraded or become unavailable. Also, by deploying network services agents

or Web monitors to determine if listeners are still responding, administrators can find root-cause failures.

Business transactions are the final product that an application delivers – the ability for a user to execute transactions. In a real-world e-business application, the user may want to log on, search for an item, and purchase that item. Each of these three steps or a combination of these steps can be modeled with Foglight.

The next section discusses how Foglight’s approach to application availability monitoring and service level policies is implemented using *application maps*.

The Foglight Approach to Availability, Performance and Service Levels

Application Maps and Service Level Policy

Foglight’s *application maps* allow users to define and organize a server population into manageable subsets by geography, department, or any other user-defined criteria. A map is a logical view of the servers and agents appearing in the master map (called the IP Map) and does not impose any extra overhead to the monitored server or to the Foglight server. A map is associated with a set of servers, monitoring agents, service level policies, rules, events, and user security.

While users can define their own maps, Foglight application cartridges – Siebel eBusiness Applications, PeopleSoft, mySAP.com and WebLogic – automatically generate their own application maps. At installation, each cartridge discovers its application environment and automatically generates a map.

Illustration 2 shows a user-defined application map named U.S. with a high level node of U.S and subsidiary nodes of Boston and Houston. Under Boston, the server troff is defined, and under Houston, sub-nodes for Development and QA, and servers houdevsap1 and houqasun1, respectively. Note that the event console on the bottom of the panel shows events associated with the selected node. In this case, the highest-level node U.S. is selected so all events pertaining to the application map are shown.

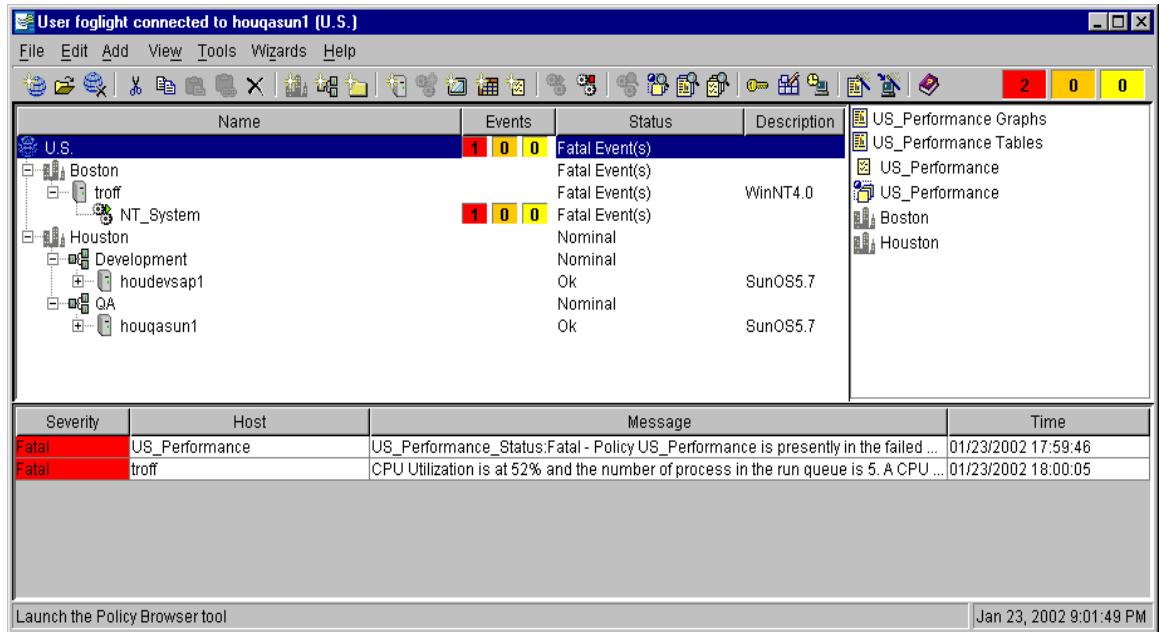


Illustration 2. Foglight Operations Console showing user-defined application map, U.S. The event console shows two fatal events, one for US_Performance, a service level violation, and another for troff, a rule violation. Service level violations are discussed below. The Foglight Operations Console is a Java application.

Illustration 3 shows the Foglight Web console, an alternative way for users to view application maps, events and other information, using a browser.

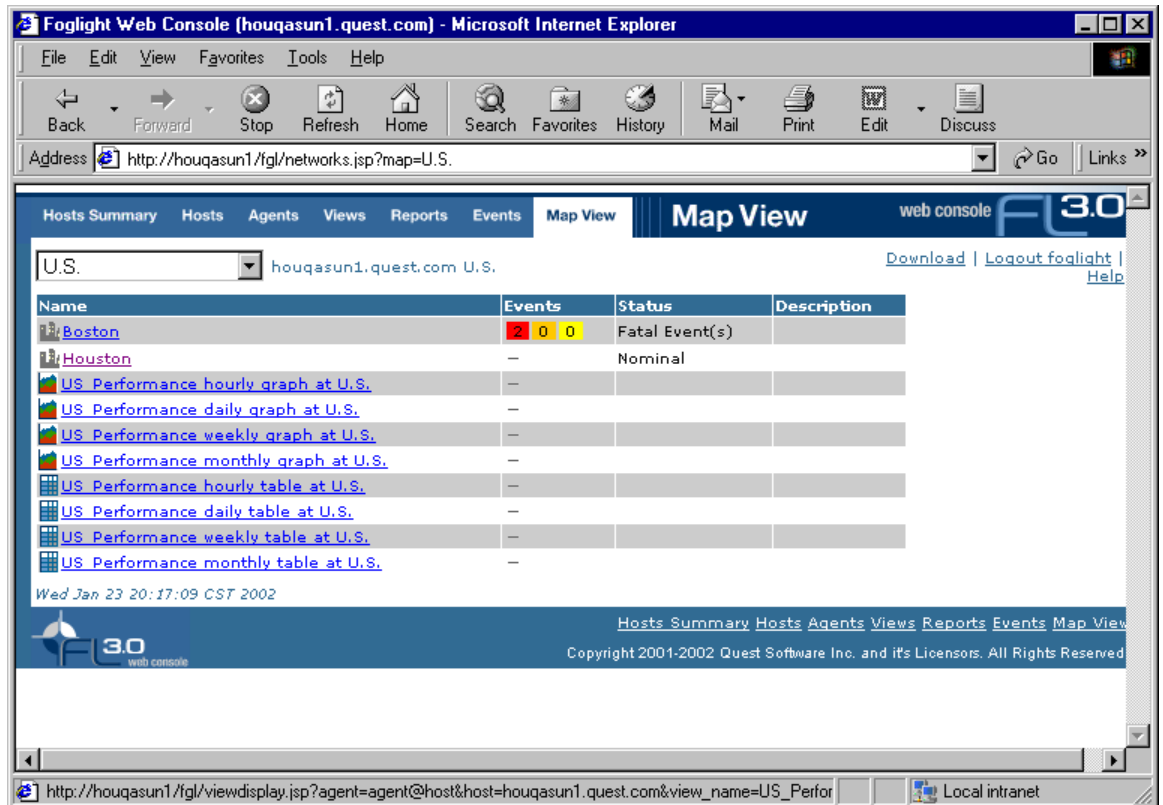


Illustration 3. Foglight Web Console showing user-defined application map, U.S.

The application map is created using a simple point and click interface. Application maps enable various professionals to see only parts of the infrastructure in which they are interested.

Service Level Policy (SLP)

The Service Level Policy facility enables users to define a service level agreement (SLA) policy that monitors and reports fatal violations for any application. SLPs can be defined using a single fatal rule or a set of correlated fatal rules covering disparate servers and agents from the application stack. SLP will record an SLA violation if *any* of the rules within an SLP definition are violated. If a violation occurs, the SLA facility will simply increment the SLP's violation counter. A violation will display on the event console, as shown in Illustration 2, and be stored for service level reporting.

Traditional SLAs are based on the availability (presence or absence) of processes making up Web, application and database servers. So, at the end of an hour, day,

week, or month, IT could deliver an SLA report showing that Web servers were available 100%, application servers (95.2%), and database servers (98.9%).

But it was difficult to *correlate* this information and show the availability percentage for all three servers. After all, if any one of these servers was unavailable, the entire application was unavailable. Moreover, if there were extensions to the application stack such as multiple application servers for load balancing, the SLA might report an application server violation *even if another application server were available* to process application transactions.

Multiple SLAs can be defined for an application, for example:

- An SLA for PeopleSoft to report on the availability for the *entire application*, defined as the availability of *all* Web, application, and database server processes, batch scheduler processes, and Citrix servers
- An SLA for Siebel batch processing to show the availability of Siebel EIM Components
- An SLA for Web administrators to report availability for all Web servers, processes and load balancing servers by regional computer center

An SLP Example

To illustrate the concept of SLP, refer to the illustration below. Assume the IT organization adopts a service level policy covering performance levels of selected U.S.-based servers and that a performance violation is defined as CPU utilization that exceeds a pre-defined threshold. For example, if server troff, houqasun1 *or* houdevsap1 exceeds a pre-defined CPU threshold, an SLP violation will be recorded.

Using the Policy Editor, an administrator creates an SLP policy named US_Performance, consisting of two nodes, Boston and Houston, and servers and agents within each node. The Policy Editor will display all the Fatal rules defined for each agent. The administrator checks the desired rules and also indicates whether the rules should be ANDed or ORed using the AND/OR menu icons.

In our example, the rules have been ORed such that if the CPU utilization on troff, houqasun1 *or* houdevsap1 is violated, an SLP violation will be recorded. However, the SLP could have been defined by ANDing the policy such that all three servers would have to exceed a pre-defined fatal rule threshold for a violation to occur.

Foglight agent rules are defined out-of-the-box for warning, serious, critical and fatal states. But only fatal rules can be defined within an SLP.

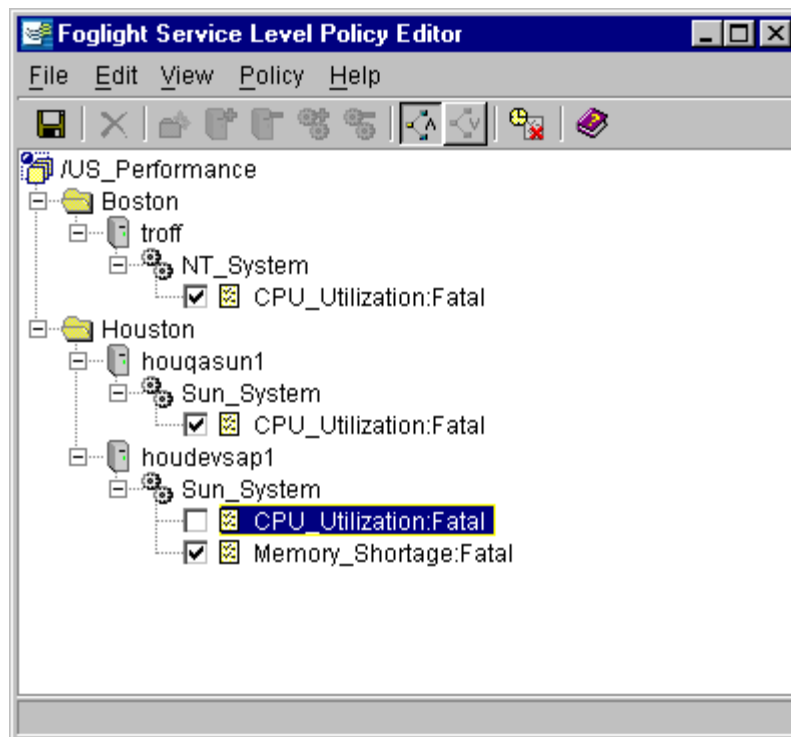


Illustration 4. Foglight Service Level Policy Map showing “US_Performance”.

The SLP engine will continually monitor the above fatal rules for violations. If an SLP violation occurs, it will display in the Foglight Operations Console event console (see Illustration 2) and be used in the SLP report (see Illustration 5).

SLP Reports

An SLP report for “US_Performance” is shown below showing 100% SLP compliance between 06:00 and 18:00 but only 6.2% compliance between 18:00 and 19:00. That is, the SLP was violated 93.8% (100.0 – 6.2 %) of the time during the interval. Availability is calculated as the percentage of total uptime for a period divided by the period.

Optionally, blackouts can be scheduled using a Foglight facility to account for scheduled downtime and maintenance periods to ensure availability is calculated within the context of scheduled application time only.

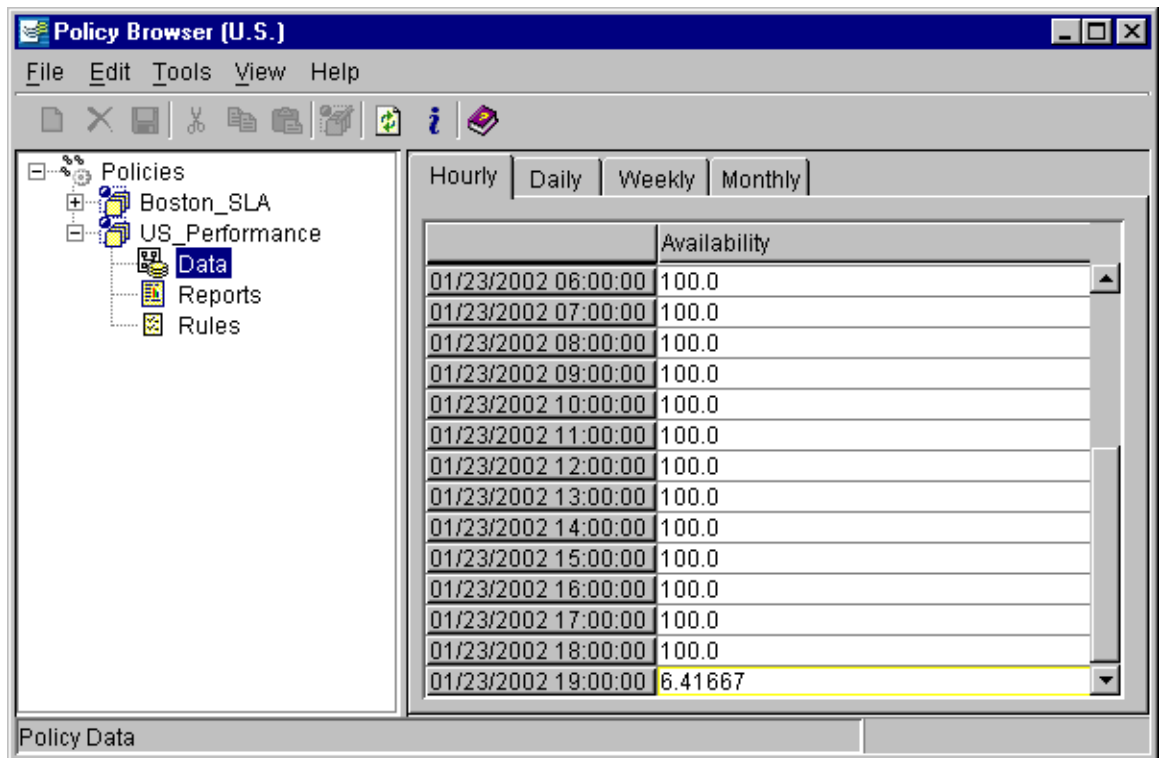


Illustration 5. Policy Map showing an hourly SLP report for “US_Performance”. Daily, weekly and monthly reports are available with a single click. Advanced features provide additional statistics such as mean time between outages and mean time to recover.

Foglight Architecture, Features and Differentiators

Foglight is a rules-based event and application monitoring solution. It provides out-of-the-box and custom solutions to bring immediate ROI in the form of event management, service level management, and capacity reporting. Foglight covers all layers of the application technology stack including network, OS, Web, middleware, database, file, SQL, applications and end-user response for Solaris, HPUX, AIX, Linux and Windows NT/W2K operating systems.

General Architecture

- **Centralized Architecture.** Designed for an n-tier heterogeneous Unix-Linux-Windows infrastructure, Foglight collects information on monitored servers using *lightweight* monitoring agents at user-defined intervals. Foglight agents consume very low CPU and memory. Foglight will not compete against your applications for CPU, memory and disk space.
- **Foglight Operations Console.** A Windows-like operations console streamlines administration and access to availability, performance, and event information. It is instrumented with navigational shortcuts, such as right clicking on an event to display an editable rule that triggered that event, or double clicking on a graph to display underlying raw information.

- **Foglight Web Console.** View Foglight availability and performance information from any browser including event notifications, server availability, agent data and rules, and reports.
- **Foglight Server.** The Foglight Server can be deployed on either a Windows or Solaris platform.
- **Foglight Installation and Deployment.** The Foglight 3 Server installs even faster than before, within minutes, and new monitored servers and agents can be easily deployed using a Wizard and point-and-click facilities.

Monitoring The Entire Application Stack

- **Platform Coverage.** Foglight monitors AIX, HP-UX, Linux, Solaris, and Windows.
- **Foundation Agents.** Foglight agents monitor all technology layers including network, network services (ftp, nslookup, ldap, database listeners, and so on), operating system, disk, file system, Web, log files, terminal services (Windows only), and system logs. Plus, the AppMonitor and Log Filter agents can be configured to monitor availability, CPU and memory of any application.
- **Application Cartridges.** Out-of-the-box and in-depth agents for Siebel eBusiness Applications, PeopleSoft, mySAP.com, Web Logic, Oracle, SQL Server, Sybase, and soon, DB2 and Oracle E-Business Suite.

End-User Application Availability and Performance

- **RTM (Response Time Measurement).** Allows users to record and playback Web-based transactions from a single or multiple network locations to simulate user experience to determine actual transaction availability.
- **SiebelBot and PeopleBot.** Provides user response time simulation for Windows client-based ERP/CRM transactions.
- **Network Services.** Provides availability and response time of pre-defined or user-defined network services such as ftp, telnet, nslookup, LDAP, and so on.

Advanced Features

- **Maps and User Security.** Application Maps enable users to define and organize a server population into manageable subsets of maps by geography, department, or any other user-defined criteria. A map is associated with a set of servers, monitoring agents, service level policies, rules, events, and user security.
- **Rules Correlation.** Enables users to express a rule based on multiple servers or agents.
- **Blackouts.** Supports blackout period(s), allowing users to disable event notification for planned “blackout” periods to accommodate monthly maintenance or other scheduled downtimes.

- **Encryption.** Optionally enables encryption of data communications as well as the Foglight repository.
- **Service Level Policy.** Powerful and innovative facility that lets users define an SLA policy that monitors, correlates and reports fatal violations for any application using any user-defined business calendar.
- **Report Wizard.** A greatly expanded reporting facility lets users easily define and schedule simple and complex availability, performance and capacity reports using an out-of-the-box facility. No 3rd party tools are required.
- **Multiple Security Levels.** Administrators can define security authorization levels by application maps ranging from read-only to complete management control for the Foglight Operations and Web consoles.
- **Scheduler.** Schedule reports, agent startup/shutdown, and blackout periods.
- **ODBC Support.** Allows users to generate SQL queries against the Foglight repository using a 3rd party ODBC-compliant application (e.g. Excel).
- **Built-In Framework Integration.** Built-in SNMP integration for selected enterprise management framework systems.
- **Online Context-Specific Help.** Detailed, context-specific documentation is an F1 key away.
- **Power User-Enablers.** The new Foglight registry and an expanded command line facility give online and scripting capabilities to configure, deploy and manage environments efficiently.
- **Seamless Spotlight Integration.** Context-specific launching of the popular Spotlight line of diagnostic products using a “Diagnose” function.

Product Differentiators

- **Foglight agents cover the entire application technology stack.** Other application monitoring products focus on a specific layer of the application stack such as the database or network, but Foglight provides coverage for multiple technology layers across heterogeneous Unix, Linux and Windows environments.
- **Quick time to value.** Foglight can be rapidly installed, configured and deployed. It also contains built-in features to make the system virtually self-administering such as automated purging and backups.
- **Foglight is self-contained.** Foglight contains monitoring, presentation, alerting, history, service level and reporting tools in *one* repository. And no 3rd party tools are required.
- **Lightweight agents.** Foglight’s agents run lighter on monitored servers for these reasons: Foglight does not store historical data on monitored servers; Foglight does not run rules on each monitored server; and Foglight agents are compiled code, not interpretive scripts. The result is a lightweight agent that provides in-depth coverage without users having to worry about monitored server CPU, memory and disk resources.

- **Correlation.** Foglight's correlation capabilities help users in the following ways:
 - Application Map.** Consolidates logical parts of an n-tier application into a single map, providing instant visibility to an application's overall health, performance, events, rules, reports, and service level policies
 - Event Management:** Define simple or correlated, cross-server and cross-agent rules
 - Drill-Down:** Drill down from high to low-level metric information with a single click, as for example, to drill down from system level CPU or memory to the associated processes and users consuming those resources
 - Service Level Policy:** Define a service level policy by correlating agents and rules from the application technology stack
- **Extensibility.** Foglight's **open agent architecture** allows easy integration of legacy scripts/programs into our centralized system. Other systems require use of proprietary languages to develop customized scripts whereas Foglight allows you to use C, C++, Perl Scripts, Unix scripts and other languages.

Summary

It is the responsibility of IT to implement the methodologies and measurements that show whether an application is available and performing to agreed-upon levels. Foglight will help IT organizations meet these objectives by addressing these important application management issues:

The Application Technology Stack. Covers the entire application technology stack and logically correlates the all the parts of an application to determine application's availability.

End-User Transaction Availability and Performance. Provides agents to monitor end-user availability and performance for Internet and Windows-based transactions.

Application Knowledge. Provides out-of-the-box agents for in-depth application monitoring for Siebel, PeopleSoft, mySAP, WebLogic, WebSphere, Oracle E-Business Suite, MS/SQLserver, Oracle, Sybase and DB2.

24x7 Availability. Reliably monitors servers and applications around the clock, locally and remotely, within and beyond the firewall. Blackout periods and report scheduling features are provided.

Root Cause Analysis. Allows drilldown into all technology layers to perform root cause analysis for any time period.



Application Service Levels and Return On Investment. Quick deployment allows IT to provide a better quality experience for users. Foglight's out-of-the-box SLA and trending reporting save IT labor-intensive report preparation.

If you would like more information about Foglight, check our Web site at www.quest.com or contact your Quest Software sales executive.

About the Author

Dennis Hejna is a product manager for application management products at Quest Software. He is responsible for the development and product direction of Foglight, an application monitoring product. Prior to coming to Quest, Dennis was a product manager with Foglight Software and a development manager with PeopleSoft in the PeopleTools group for several years.

About Quest Software

Quest Software, Inc. is a leading provider of application management solutions. Our products increase the performance and uptime of business-critical applications while driving down the total cost associated with running those applications. By focusing on both the people and technology that make applications run, Quest Software enables IT professionals to achieve more with fewer resources. Quest Software is based in Irvine, California and has offices around the globe. For more information, visit www.quest.com.

For more information on Quest or Foglight, contact:

Dennis Hejna, Product Manager
Quest Software, Inc. at (925) 737-4043
Dennis.Hejna@quest.com

Last Updated: 3/13/2003