

PRODUCT BRIEF

Altaworks
Commander™

M

Measure

Isolate

Resolve

Altaworks Application Quality Management (AQM) Solutions reduce the cost of resolving Application Performance problems throughout the life cycle.

Target Solutions

- Application Service Level Monitoring
- Performance Analysis in Production
- Performance Analysis Prior to Deployment

KEY CAPABILITIES

Services Modeling

- Assure end-user and business service levels
- Monitor real-time and historical service level compliance
- Increase visibility and management of application infrastructures
- Quickly identify performance problems across distributed, heterogeneous application environments
- Respond immediately to performance problems before they impact end-users or the business

Agentless Testing

- Assess end-user experience from inside or outside the firewall
- Test critical infrastructure components behind the firewall
- Quickly isolate problems through test triangulation
- Calculate relevant service level metrics such as response and connection times

WHY ALTAWORKS?

Commander's deployment simplicity, service object model, and ease-of-use make it easy to be monitoring complex business-critical applications within a few hours of initial installation.

Altaworks Commander is the robust, cost-effective early warning system that combines end-user availability and performance monitoring with day-to-day infrastructure operations management. It quickly answers these crucial questions for the most complex environment:

- Are critical applications up and running?
- What components are slow? Which have failed?
- Is availability and response time compliant with Service Level Agreements?

Commander employs an outside-in approach to problem identification and isolation. It reaches beyond ad-hoc component monitoring to capture transaction-level, end-user experience.

Applications today rely on a tiered infrastructure to deliver a business service through transactions. Each transaction traverses firewalls, network devices, web servers, application servers and databases. In many cases redundant, load-balanced resources are used to ensure availability and improve performance.

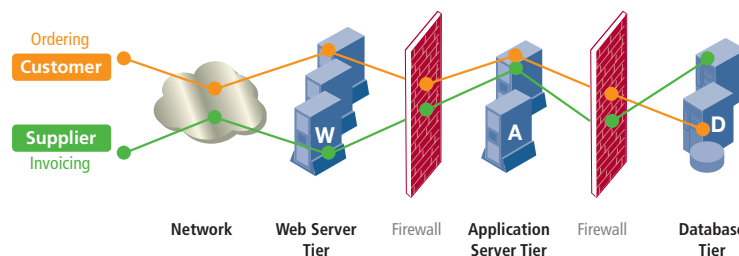


Figure 1. User transactions traverse multiple components across tiers..

Service Model Encapsulates Both End-User and Infrastructure Monitoring

Key to Commander's effectiveness is its patent-pending Service Model approach to representing availability and performance. The Service Model works across application tiers so you can easily determine where a problem is occurring. More importantly, the Service Model lets IT organizations combine end-user perspectives with infrastructure management.

This unique approach enables IT staff to keep its focus on the end-user experience when determining specific problem priorities and overall business impact. Service Models are comprised of the following components:

- A **Service** represents an application or any high-level business function with which users interact. A service is made up of all of the interdependent components that are needed to complete a transaction. These interdependent components—whether physical or virtual—are organized by tiers and elements.
- A **Tier** is any logical grouping of elements. Commonly, a tier contains elements with similar features and functions. It can also represent clustered or redundant components. For example, in a banking Service a typical tier may be comprised of all the web servers for online banking.

- An **Element** represents an individual component needed for a service. An Element can be an application, operating system, database, web server, firewall, router, etc.

For every element, tier, and service in the Service Model, Commander can run a wide variety of tests that generate availability and response time data for any infrastructure component or application in your environment. These tests automate the same steps that an operator would take to validate that an application is available and responding within defined limits.



Figure 2. Commander's dashboard display.
Gives you a constant update on the availability and response time for critical Service Level Objectives (SLOs).

Distributed, "Agentless" Architecture

Commander's distributed architecture allows it to encompass the most complex infrastructure. Commander's agentless approach requires no installation on systems that you want to monitor.

- The Controller is installed on a single system and is responsible for receiving, processing, and storing data from all Commander tests, as well as providing the user interface.
- The Central Test Engine is part of the Controller installation and resides on the same system as the controller. It runs tests to establish an internal 'best-case scenario' baseline.
- Remote Test Engines can be installed on any system from which you want to measure end user transaction availability and performance. Test Engines can be deployed at various physical locations inside and outside a firewall for close simulation of end-user traffic, for example at different data centers or at key partner sites.

Tests can be scheduled to run on a recurring schedule or on a periodic interval. For example, it is common to run regular web content verification tests against critical content on 1-5 minute intervals.

Services, the Test Engine and Service Level Objectives

Service Level Objectives (SLOs) provide the ability to monitor the end-user experience for critical indicators of service level performance by linking performance or availability limits for a specific test to a particular service.

For example, the "Online Sales" service might have an SLO that specifies that the service must have a user response time of less than 5 seconds, 90% of time, reported daily. The Commander Test Engine monitors these requirements by running tests that emulate the actions of user transactions, whether they are based on HTTP or other TCP protocols. Commander updates SLO status in a special dashboard display and in the main user interface and generates historical reports on demand.

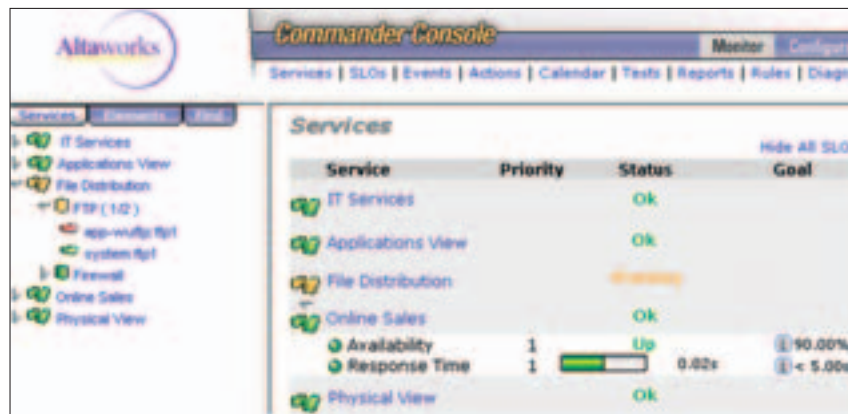


Figure 3. Commander's user interface shows the Service Model as an expandable hierarchy. You can quickly identify availability or performance issues, and isolating a problem is as simple as clicking on the problem Service, Tier and Element.

PROVEN ROI

With Commander you can deploy quickly and simply and begin enhancing IT operations staff productivity almost immediately. In addition, the cost to get started is significantly lower than competing products.

CALL FOR A FREE DEMONSTRATION

Get an online demo of Commander along with an initial assessment of how Altaworks can help you implement an early warning system for your applications. Call us at 888-877-7111 or write at sales@altaworks.com.



MINIMUM SYSTEM REQUIREMENTS

The Commander Console/Test Engine requires a Windows 2000 Server machine with Oracle Enterprise Edition V8.1.7.

The Test Engine runs on Windows and Solaris systems. The Remote Test Engine requires minimal system resources on existing Windows and Solaris systems.

COMMANDER TEST TYPES

- Basic HTTP
- Multiple URL
- Transactional URL
- Custom Script
- File Content Validation
- DNS
- FTP
- IMAP/SMTP/POP
- PING
- Trace Route
- SNMP
- TCP Port Check
- Transactional Generic TCP



Altaworks Corporation
55 Northeastern Boulevard
Nashua, NH 03062 USA
Ph. (603) 598-2582
www.altaworks.com