



JaxView Enterprise Edition

SOA Management and Runtime Governance

JaxView is a monitoring and runtime management tool designed to meet the challenges of operating and managing service-oriented infrastructures. Built as a platform-independent Java application, JaxView is designed for ease of installation and administration. JaxView manages Web services environments by accessing the XML-based messages exchanged between service providers and consumers. Its versatile deployment options can virtualize service endpoints, enable auto-discovery of Web services, integrate with ESB's, or use distributed message relay agents. When deployed as an XML firewall, JaxView enables versatile policy enforcement capabilities including client authentication, XML message security options, and WS-Policy enforcement. By accessing the XML service messages, JaxView can provide more meaningful data about service performance, client usage levels, and policy compliance than end-user or server log monitoring tools. With a rich set of features, JaxView is a cost-effective management solution for service-oriented architectures.

Key Functionality

Web Services Monitoring and Management

JaxView uses objects called monitors to record and evaluate copies of Web service messages for metrics relating to service quality, performance, and usage. Copies of request and response messages are captured without impacting service performance.

Performance Monitoring:

- Service Usage Rate
- Service Response Time
- Service Throughput
- Message Size

Service Fault Monitoring:

- Fault rate per time period
- Fault totals per time period
- Monitoring for specific fault code or fault string

Client Metering and SLA Monitoring:

- Client Usage Rate
- Per client throughput
- Faults per client
- SLA events and reporting

Active Monitoring:

- Monitor service availability with periodic synthetic requests
- Includes active monitoring of multi-step transactions

Transaction Monitoring:

- Monitor generic transactions that implement WS-Transaction and WS-Coordination standards
- Step-wise performance for each transaction instance
- Monitor the number of commits or rollbacks for transactions

Event Notification and Alerting:

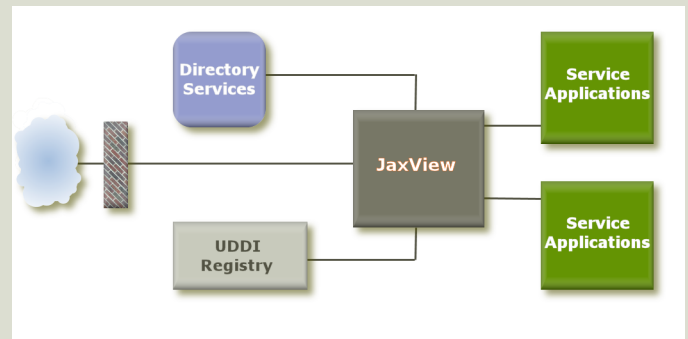
Performance and service quality event notifications are generated when a performance threshold has been exceeded. The alerting media include:

- Email
- JMS
- Script Execution
- SOAP
- SNMP

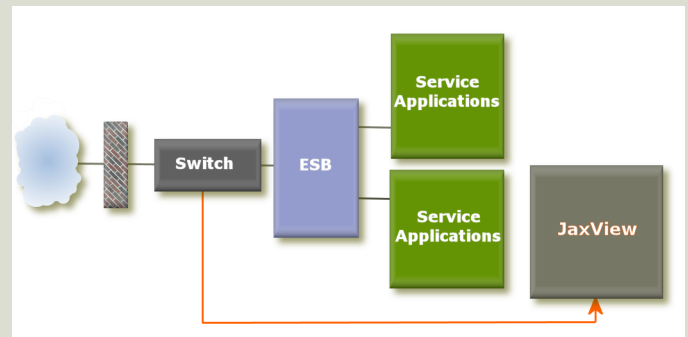
Deployment Options

Each installation of JaxView allows for different deployment configurations to meet operational objectives and requirements.

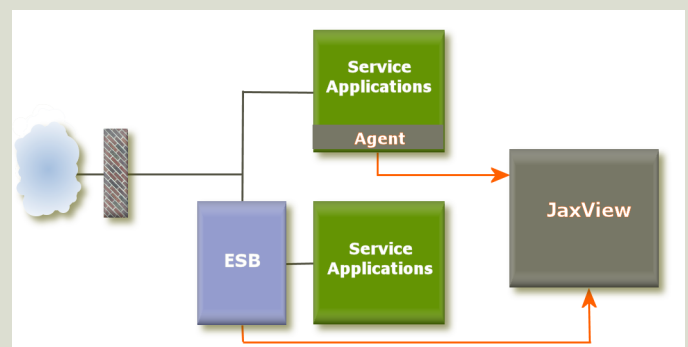
1. Gateway/XML Firewall - Deployed between Web services consumers and providers to enable runtime policy enforcement and Web services security options.



2. Network Appliance - Installed with a network sniffer to passively listen for Web service message TCP packets (including SSL). Allows for auto-discovery of Web services.



3. Agent Architecture - Agent modules are installed on J2EE and .NET application servers to replicate Web services messages to the JaxView server.
4. ESB/MQ Integration - JaxView can retrieve copies of service messages by subscribing to ESB products that use Java Message Service (JMS).



Event notification thresholds include options for calculating statistical metric baselines to reduce unnecessary alerting. Options include:

- Rolling Baseline - Averages recent metrics and generates the standard deviation based on a selected time interval.
- Date Specific Baseline - Calculates a baseline relative to metrics recorded on a specific baseline date.
- Weekly or Daily Baseline - Combined with a Rolling Baseline, calculates a baseline to include week-over-week or day-over-day performance trends.

Reporting

Includes performance reports, per client SLA reports and audit reports. Reports can be generated on-demand or on a set schedule. Reports are saved for later reference and can be forwarded by email.

Security Policy Enforcement

When deployed as a gateway or XML firewall, JaxView can provide centralized enforcement of security policies for Web services. Supported technologies include:

- LDAP / Active Directory authentication
- XML Encryption
- Digital certificates
- WS-Security
- STS integration
- WS-Trust
- X.509 authentication.
- SAML authentication

Message Brokering

When deployed as a gateway or XML firewall, JaxView can perform certain message brokering functions.

Request/Response Modification

Request or response messages can be modified in runtime before being forwarded to their destinations.

Throttling

Limits can be set to control the number of requests per hour or per day that JaxView will forward to a Web service.

Load Balancing and Failover

JaxView can load balance requests between multiple endpoints. Alternately, if it detects a primary endpoint is down, requests can be redirected to a failover service endpoint.

Content-based Routing

Messages can be routed to different endpoints based on the content in the message.

SOAP to JMS Conversion

Incoming HTTP SOAP messages can be converted to the JMS protocol and forwarded to a service bus or message queue.

System Requirements

Win 2000/XP/2003 or Linux or Mac OS X
 500 MHz + processor
 20 Gb+ Hard disk space
 Internet Explorer 6+ or FireFox 1.5+ (ActiveX enabled)

Integration Options

UDDI Registries

Bi-directional integration with UDDI registries allows JaxView to be automatically updated from a central management registry or to push policy updates to a registry.

External Databases

By default JaxView uses the file system for data persistence but it can store data and update configuration settings from a shared external database.

ESM Consoles

Events can be forwarded to ESM consoles as SNMP traps, SOAP messages, or emails.

Web service API

JaxView can be configured and report data using a Web services interface.

Customization Options

JaxView has an open API for customizing many different aspects of the application itself. These include:

- Web Service configuration API for controlling the JaxView application itself using SOAP messaging
- Monitor API for creating custom monitor types
- Alerting API for creating custom alerting mechanisms
- Rule Engine API for creating custom event rules
- Logging API for exporting service metrics to other reporting tools or enterprise system management consoles.

Supported Technologies

Programming/Development Environments:

- J2EE Framework 1.3.x, 1.4.x
- .NET Framework

Supported Protocols:

- HTTP/HTTPS
- JMS
- .NET Remoting
- JAX-RPC

Supported Web Services Standards:

- SOAP 1.1, 1.2
- WSDL 1.1
- UDDI
- WS-Security
- WS-Trust
- BPEL
- WS-Coordination
- WS-Transaction
- WS-Policy

Supported Application Servers:

- Apache Tomcat 4.x
- SunOne App Server 8.x
- WebLogic 7.x, 8.x
- Apache Axis
- MS-IIS .NET
- WebSphere 5.x

Supported Databases

- MySQL
- DB2
- Oracle
- MS SQL Server