



The JBoss Network™

IMPROVING OPERATIONAL EFFICIENCY



Contents

THE JBOSS NETWORK™ - IMPROVING OPERATIONAL EFFICIENCY

| | |
|---|----|
| Executive Summary | 1 |
| JBoss Network Customer Portal – Overview | 2 |
| Case Management Module | 2 |
| Knowledge Module | 3 |
| Registration Module | 4 |
| Download Module | 4 |
| JBoss Network Enterprise Manager - Overview | 5 |
| JBoss Network Enterprise Manager – Architecture | 7 |
| Architecture Overview | 7 |
| Server Architecture | 7 |
| Agent Architecture | 8 |
| Inventory Module | 10 |
| Administration Module | 11 |
| Monitoring Module | 12 |
| Update Module | 13 |
| Provisioning Module | 13 |
| Summary | 14 |
| | |
| Appendix I - JBoss Network Customer Portal Features | 15 |
| Appendix II - JBoss Network Enterprise Manager Features | 16 |
| Appendix III - JBoss Network Enterprise Manager – System Requirements | 17 |
| Appendix IV - Monitoring Module Detailed Feature List | 18 |
| Appendix V - JBoss Application Server Metrics | 19 |
| Appendix VI - Tomcat Monitoring Metrics | 20 |

EXECUTIVE SUMMARY

The JBoss Network™ is a major component of the JBoss Subscription: a set of services and tools that assist enterprises during every stage of their application lifecycle – from design and development through testing and deployment to on-going management and monitoring of their applications. The combination of enterprise-class Middleware software offered with no-cost open source licenses, the highest rated technical support services in the industry, and JBoss Network operational tools have led JEMS (JBoss Enterprise Middleware System) products to be the most widely deployed Middleware products in the industry today.

JBoss Subscriptions allow you to:

Reduce technical, business, and legal risks by relying on JBoss for certified software, patches and upgrades; accurate technical knowledge; and advanced management tools for preventing service interruptions.

Save time and money by relying on JBoss to help you avoid design and development mistakes, be alerted to potential issues before they cause service interruptions, and more quickly repair production faults if they occur. Your staff can also save a significant amount of time currently spent on the JBoss website by subscribing to patch and software upgrade and technical content notifications.

Improve Operational Efficiency by increasing your visibility over your entire JEMS Middleware environment, more effectively managing and rolling out patch and software upgrades, monitoring all key resources, and keeping an accurate audit trail over your entire process.

The main components of a JBoss Subscription are:

Enterprise Support Services – consisting of expert technical support, developer-to-developer assistance, and bug and patch support. All support cases are handled by Java experts – right from the very first call. 24x7 support with up to two hour response times are available. The Developer Assist offering consists of expert advice that spans architecture, design, configuration, optimization, and turning recommendations. It's like adding a team of subject matter experts to your development organization. The bug and patch support offers unparalleled turnaround times for bug-fixes and certified, binary patches.

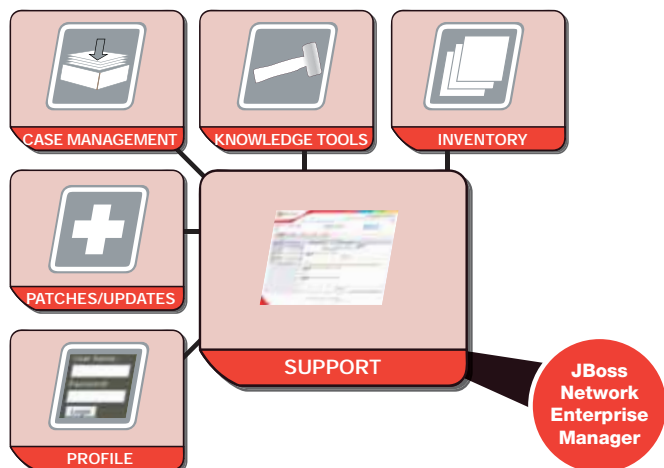
JBoss Network Customer Portal – an online self-service portal that connects subscribers directly to the expert support staff at JBoss. Consists of four primary modules: Knowledge Module for quickly accessing accurate technical knowledge across all JBoss website data sources; Registration Module for registering your JEMS resources and subscribing to patch and knowledge alerts; Download Module for obtaining the latest certified patches and upgrades; and Case Management Module for opening and tracking support cases.

JBoss Network Enterprise Manager – platform for managing and monitoring enterprise JEMS deployments. At the time of this writing, there are three primary modules: Inventory Module control for identifying and cataloging your Middleware assets; Administration Module for performing functions such as start, stop, and re-start on single applications or across an enterprise cluster; and an optional Monitoring Module for monitoring the entire JEMS platform including Apache Tomcat along with Apache Web Server and leading operating systems. In addition, Software Update and Provisioning Modules will be released in early 2006.

This paper will cover the JBoss Network Customer Portal and JBoss Network Enterprise Manager components in detail. For more information on the JBoss Subscription, please visit www.jboss.com/services/index.

JBoss Network Customer Portal – Overview

The JBoss Network Customer Portal is a Web-accessible portal that connects enterprises with JBoss Subscriptions to the JBoss



Support Team, Certified JBoss Patches and Upgrades, and accurate technical knowledge useful for solving technical issues. The Portal is generally divided into logical tabs or modules. Each of the four main modules of the JBoss Network Customer Portal offers unique functionality as detailed over the next few pages.

The Customer Portal connects you directly to the JBoss Customer Support team, certified patches and upgrades, and accurate technical content.

Case Management Module

The case management module allows subscribers to open, track, and close support cases. Although cases can be opened through other mechanisms (i.e. via telephone); the Customer Portal is the most often used window into the JBoss support staff. Support cases can span from architectural design questions to development suggestions to configuration and optimization issues to bug fix requests in production. JBoss Subscriptions are available with various service level agreements ranging from two business day response times up to 24x7 with two hour response times.

When opening a support case with the case management module, subscribers will be presented with immediate potential solutions via a unique metadata-based search capability embedded within the application. Subscribers will still be contacted directly by the JBoss support team; the potential solution information is simply additional, immediate data that can potentially assist in resolving the issue in a timely manner.

The case management module also provides useful historical information about current and closed support cases. This is especially useful for auditing purposes.



The Case Management Module is used to open, edit, track, and close support cases with the JBoss Customer Support team.

Knowledge Module

The knowledge module is an extremely popular component of the JBoss Network Customer Portal primarily because it saves organizations significant amounts of time locating relevant technical knowledge across the expansive JBoss set of data sources. This module provides you one-click search capabilities across a wide variety of data sources including:

- » Wiki knowledgebase
- » Forums
- » Documentation
- » Javadocs
- » JIRA bug tracking system
- » Source code
- » Subscriber's existing and previous support cases (coming in Q4 2005)
- » Premium content available only to JBoss Subscribers

Premium content includes items such as architectural guides, performance tuning guides, migration tools and guides, and other useful documents. Related, certified solutions are made available to JBoss Subscribers and made available via the knowledge module. These solutions provide technical recommendations for overcoming common issues experienced by other subscribers. Examples include integration recommendations for third-party software, frequently asked questions about complex topics (i.e. clustering), and other popular topics (i.e. debugging JNDI, driver recommendations).

The knowledge module leverages a powerful meta-data system that ensures subscribers receive relevant and accurate information, essentially filtering out data from un-trusted sources.



The Knowledge Module provides one-click search access across all JBoss knowledge sources and ensures accuracy of the data.

Registration Module

The registration module allows you to register your Middleware assets for purposes of receiving future notifications about patches or new or updated knowledge content specific to your environment.



The Registration Module is where you register your JEMS resources and versions and subscribe to relevant software and content notifications.

You can register all JEMS Middleware and operating system resources including their version numbers. Registration can occur all the way from the server level down to a particular instance level, depending upon the granularity desired for notifications. Many subscribers also find benefit simply from the fact that they can access configuration and version information about their often complex Middleware infrastructures.

As importantly, the registration module allows you to subscribe to patch and update notifications as well as knowledge content updates. All notifications are relevant to your environment. This capability is another tremendous time saver; eliminating the need for your staff to continually scan various JBoss website sections or closely track developer mailing lists.

Download Module

At present, the download module is a repository for the latest patches and upgrades. Longer-term, the download module will be the home for all certified, production-ready versions of each and every JEMS product.

JBoss Subscribers have access to the latest patches and upgrades in binary form. Non-subscribers must manually search for code patches via the Sourceforge.net CVS repository and must compile on their own for immediate access to the patch. An additional issue for non-subscribers is that they are generally not alerted to the availability of new patches. Notifications of new relevant downloads can also be provided as discussed in the previous section. The download module will also be able to provide an audit trail of downloads (coming Q4 2005).



The Download Module is where you will find the latest certified patches and software upgrades.

JBoss NETWORK ENTERPRISE MANAGER - OVERVIEW

The JBoss Network Enterprise Manager is an advanced management platform for inventory, administration, monitoring, and (coming in early 2006) updating, and provisioning JEMS. Each module is accessible through a separate tab within the Enterprise Management console.

The JBoss Network Enterprise Manager is an agent-based platform that is deployed locally. The five above-mentioned modules are each full-featured management products all of which are integrated together into a single console. Core Enterprise Manager services including fine-grained user access control, auditing, scheduling, and alerting span all modules. There is also a tight integration between the JBoss Network Enterprise Manager and JBoss Network Customer Portal described in the previous section.

The 5 core modules of the JBoss Network Enterprise Manager are:

Inventory Module

The inventory module allows you to catalog IT assets spanning platforms (Linux, HP/UX, Solaris, Windows, AIX), servers (Apache Web Server, Apache Tomcat, JBoss AS), and services (EJB, Message Driven Beans, data sources). Assets can be added manually or auto-discovered providing enhanced visibility over your critical business applications including their versions and dependencies.

Administration Module

The Administration Module provides a single location for performing key control functions such as start, stop, and re-start across the entire JEMS platform. Administration functions can be applied to a single application or across an enterprise cluster. Operations can also be scheduled for later dates on-demand or on a recurring basis. This module also allows you to roll-back to previous versions if necessary.

Monitoring Module

The Monitoring Module is an optional component of the JBoss Network Enterprise Manager. Once purchased, all monitoring capabilities are fully integrated and accessible from the JBoss Network Enterprise Manager console.

Extensive monitoring capabilities are provided for the JEMS platform along with supporting components such as operating systems (Windows, Linux, and UNIX), Apache Tomcat, Apache Web Server, and your own JEMS-based applications. The Monitoring Module is continuously building and updating a model of behavior of every measurement being collected. This allows you to define alerts relevant to the actual behavior of your infrastructure. Baselines are also utilized to identify out of band problems. This allows you to quickly identify resources within your infrastructure that require attention as well as get a historical view of what was happening at the time of the fault.

You can create conditional alerts and define actions that should be initiated once an alert is raised. Graphing and charting including trend analysis along with a complete audit trail of alerts are also provided.

Update Module (Beta Q4 2005, Release Q1 2006)

Because updates and upgrades are always a difficult challenge to manage for enterprise software deployments, JBoss is building out a Patch and Software distribution management system that will be available through the JBoss Network Enterprise Manager. In many respects, this module will be similar to a Windows Update service for JEMS software.

Updates can be reviewed, rejected, or applied. They can be applied to individual environments in a controlled fashion allowing you to move applications from development to QA and test to production. Software updates are a schedulable task and can be done in a rolling fashion or all at once.

Provisioning Module (Q1 2006)

The Provisioning Module will help you to configure new environments and recreate existing environments quickly. The latter is especially useful for cases when you need to mirror your production environment in QA, a common requirement for many IT shops. This requirement is usually a challenge as is keeping environments synchronized. The Provisioning Module will provide you the ability to create and replicate environments. It will also allow you to easily compare and synchronize environments and provide you the ability to roll-back to previous states. An audit trail will also be provided, along with integrations into popular ticketing systems.

JBOSS NETWORK ENTERPRISE MANAGER – ARCHITECTURE

Architecture Overview

The JBoss Network Enterprise Manager is a distributed J2EE application that runs within the JBoss Application Server on a number of different operating systems including Linux, Windows, Solaris, HP/UX, and AIX. It is based on a server-agent plug-in architecture that delivers a wide range of management functions organized into logical areas or Modules. Current modules include Inventory, Administration, Monitoring, and Updates. A provisioning module will also be released later in 2005.

Agents reside on each system that is managed. Standard agents are supplied for primary JEMS infrastructure components including JBoss AS, Hibernate, and Apache Tomcat along with Apache Web Server and various operating systems. An open plug-in developer's kit is also provided allowing you to implement additional server types. Agents can run standalone, embedded, or remote thus supporting a wide range of scenarios.

The JBoss Network Enterprise Manager provides an integrated view of all modules from a single user interface. Common services including fine-grained control of user access and permissions, auditing, scheduling, and alerting are also provided across all modules.

Server Architecture

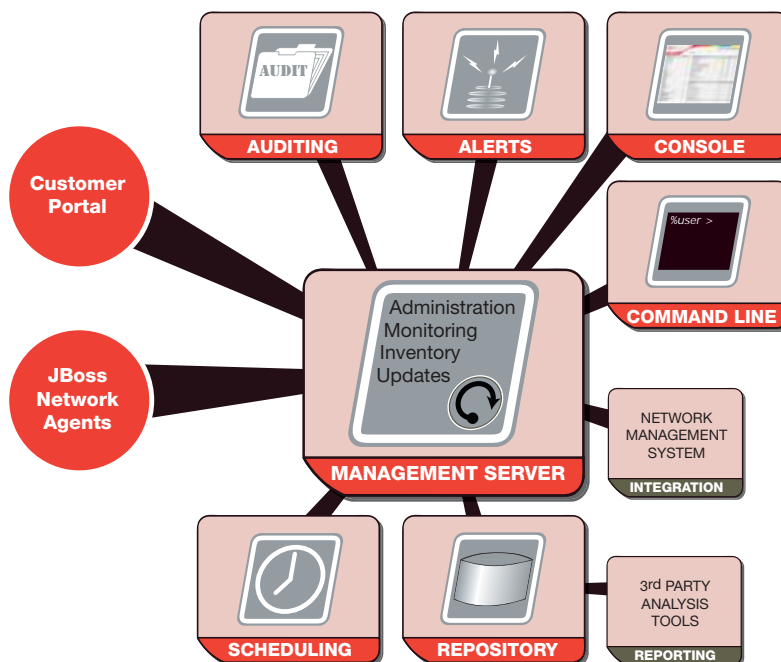
The JBoss Network Enterprise Manager is a composite of Web and J2EE applications that run within JBoss Application Server.

The agent-based implementation allows for a compact memory and CPU utilization footprint.

The server side is broken down into logical Modules; each of which will be explained in more detail later in this document.

User Interfaces - User interactions occur through two primary interfaces: the JBoss Network Enterprise Manager User Interface (UI) and a command line interface. There is also a tight integration with the JBoss Network Customer Portal (described earlier) for support and patch information. The server uses a secure socket protocol to communicate with the JBoss Network Customer Portal and all agents.

Repository – the repository is a key component of the JBoss Network Enterprise Manager. It stores up to two years of monitoring data – along with all other related data including audit trails – without any performance or scalability penalties. The built-in database provides a simple, administration-free repository that stores all data in a relational schema. Postgres ships as the default database for the repository although other databases can be used including Oracle.



The JBoss Network Enterprise Manager server is command central for managing your JEMS environment; receiving raw management data from Agents and updates from the Customer Portal.

Events and Alerting – the JBoss Network Enterprise Manager event bus serves as the central messaging bus for all monitoring, inventory, and control events. The event bus manages the triggers and actions that the JBoss Network Enterprise Manager uses to implement alerts (see Monitoring Module section for more details).

Auditing – audit trails are available across the complete set of Modules.

High Availability – high availability is achieved by clustering the JBoss Network Enterprise Manager. A high availability subsystem manages distributed object caches and ensures data consistency across the cluster using a lightweight broadcast messaging framework among all members of the cluster. Note: high availability requires an Oracle database (8 or 9).

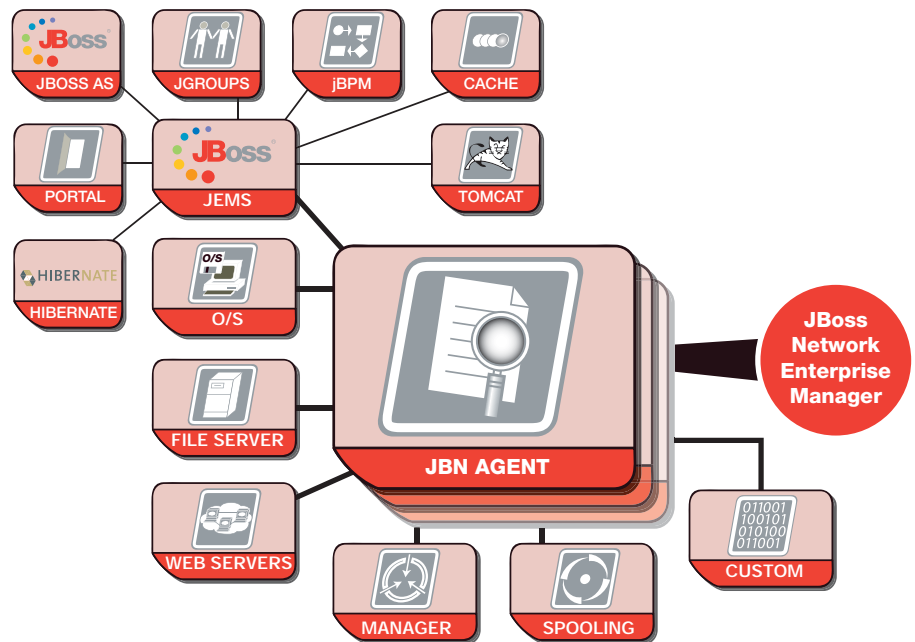
Security / Access Control – the JBoss Network Enterprise Manager authenticates user logins via an LDAP repository, Microsoft Active Directory, any SQL data store, or its own data repository and enforces the JBoss Network Enterprise Manager’s role-based system security policies. Roles give fine-grained access control. For example, a developer role might only be able to view EJBs on a particular JBoss instance; an architect might be able to view and alert in certain groups but not others; and an administrator might be allowed to perform control actions such as start/stop/re-start of applications. In addition, the Monitoring Module can also continuously scan JBoss Configuration files looking for changes in files, file sizes, or permissions; further securing your JEMS environment.

Server API – spanning across all Modules, the JBoss Network Enterprise Manager API aggregates all module and subsystem-specific functionality under one umbrella. This provides a simple programmatic interface for all JBoss Network Enterprise Manager features. Even the user interface and shell are built on top of this API.

Agent Architecture

JBoss Network Enterprise Manager agents are primarily implemented in Java and run on Linux, Windows, Solaris, HP/UX, and AIX. The agent core consists of a lightweight kernel that contains a listener and command processor. The listener implements a command protocol over SSL and is responsible for marshalling messages on and off. The command processor loads plug-ins and their interfaces onto each JEMS resource and also delegates incoming requests to the appropriate Module.

Plug-Ins – each agent has a set of plug-ins for inventory (auto-discover), administration, and monitoring. Additional plug-ins will be added for updating and provisioning in the coming months. Each plug-in has an interface that allows it to complete its desired work - for example gathering a JMX metric or re-starting a service. All plug-ins interact with the products they support via standards-based interfaces such as JMX, SNMP, and JDBC. All communications take place entirely on the local machine; eliminating the need to share sensitive data over the network.

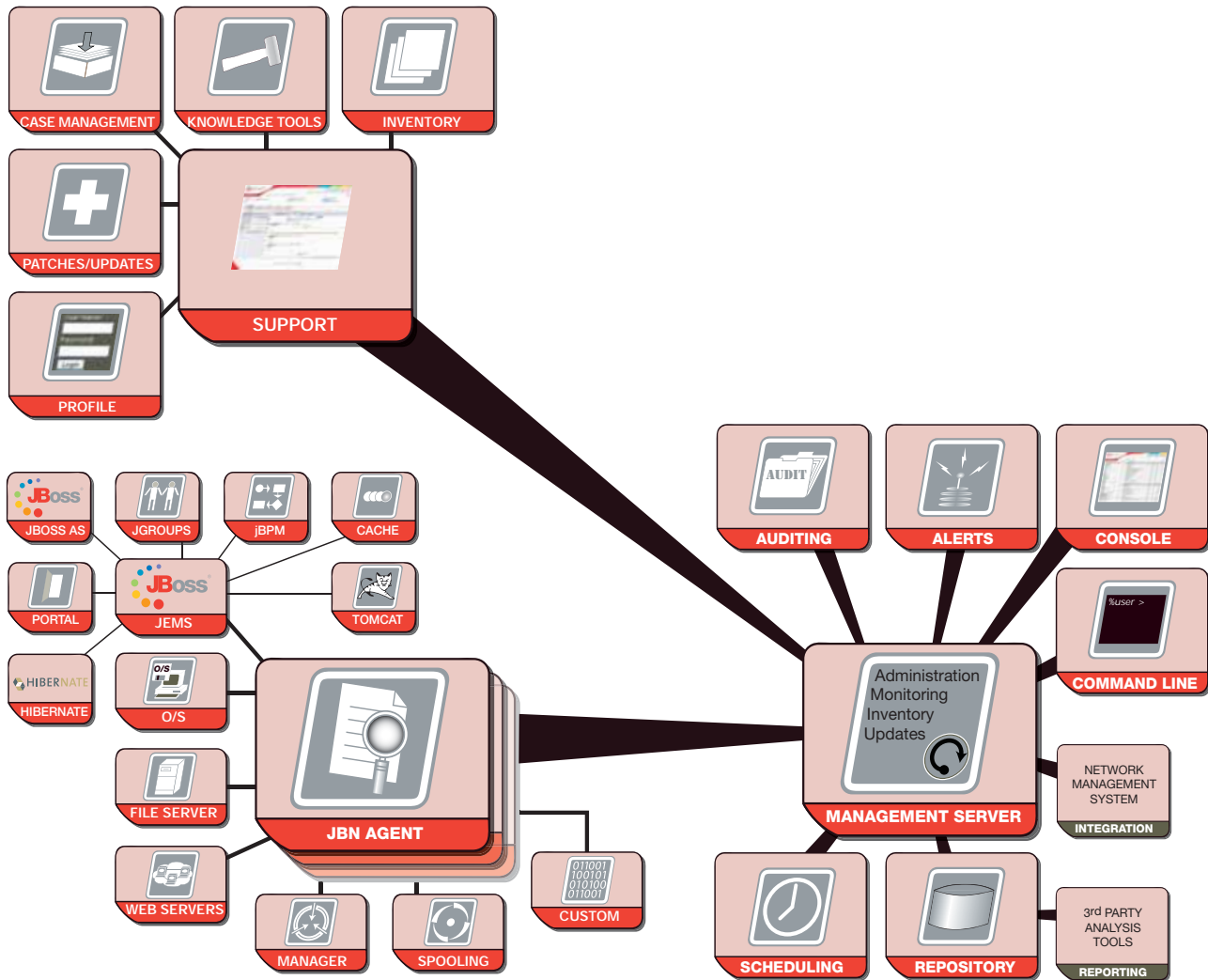


Agents and their plug-ins capture data from managed resources and send it to the JBoss Network Enterprise Manager server.

In addition to the standard JEMS plug-ins, JBoss also provides a plug-in developer's kit that allows you to develop your own custom extensions.

Fault Tolerance – even if the JBoss Network Enterprise Manager is temporarily unavailable, all agents will continue to operate normally. Captured data (i.e. monitoring metrics) will be spooled to disk. Once the JBoss Network Enterprise Manager is visible again via the network; the agent will transmit the spooled data. Additionally, the configuration data for each agent (i.e. metrics collected, control schedule) is stored centrally at the JBoss Network Enterprise Manager.

Distributed Management – the JBoss Network Enterprise Manager provides a deployer tool that greatly simplifies the process of installing agents. The deployer tool allows you to deploy agents to hundreds of machines easily; allowing you to direct large deployments from the JBoss Network Enterprise Manager console. The deployer is implemented using a platform-independent implementation of the SSH protocol thus ensuring that all JBoss Network Enterprise Manager network traffic is secure.

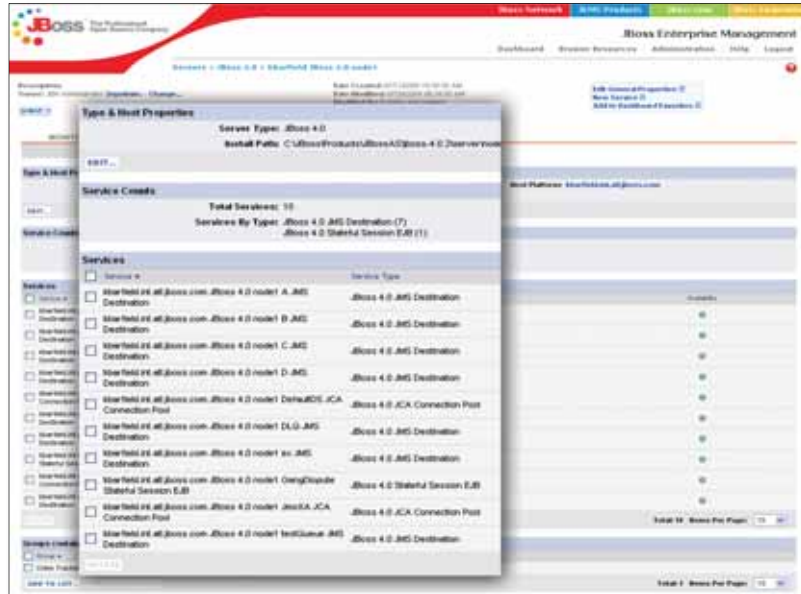


The JBoss Network is a fully integrated management and support platform for improving operational efficiency of your JEMS environment.

The following pages will describe in more detail the five primary Modules within the JBoss Network Enterprise Manager.

INVENTORY MODULE

The Inventory Module provides you a constant and current representation of your entire JEMS infrastructure, JEMS-based applications, and key dependencies. This module allows you to logically manage the topology of your JEMS Middleware assets; thus providing you enhanced visibility over your critical business applications including their versions and dependencies.



The Inventory Module allows you to easily auto-discover resources and provides a current and accurate view into your entire JEMS environment.

discovery mechanism. The auto-discovery system within the JBoss Network Enterprise Manager processes incoming auto-discovery reports from various JBoss Network agents. It also manages the approval process for merging auto-discovery data into the master JBoss Network Enterprise Manager inventory and maintains the auto-discovery scan schedule.

The physical topology includes platforms (Linux, Windows, Solaris, HP/UX, AIX), servers (Apache, Tomcat, JBoss AS), and services (EJB, MDB, data sources). The logical topology allows you to create auto-groups, mixed and like groups, clusters, and applications – each of which can be comprised of a set of platforms, servers, and services. The logical topology is an important concept as it allows you to apply a single operation across your environment or collect statistics and view them as an aggregate. It allows you to manage complete business applications instead of individual components that make up that application.

Assets can be added manually or, more popularly, added via a powerful auto-

ADMINISTRATION MODULE

Once JEMS Middleware assets are discovered and stored in the repository, they can then be managed. A key aspect of the JBoss Network Enterprise Manager is the Administration Module. The Administration Module provides a single location for performing key control functions such as start, stop, and re-start across the entire JEMS platform.

The administration module supports low level access to all configuration files – either from a single resource or across a group of resources. This allows common administration functions to be applied to a single application or across an entire enterprise cluster. All low level access is provided via common standards such as Webdav and JMX.

JBoss Network Enterprise Manager provides you the flexibility to selectively manage individual configuration files within your JEMS environment. In this scenario, the JBoss Network Enterprise Manager will identify marked configuration files and begin to version it and help to manage its changes and deployments on an on-going basis.

Operations such as start, stop, and re-start can be initiated on-demand or scheduled – either for a particular time in the future on a recurring basis. The Administration Module provides the flexibility for the operations to occur in parallel or serially (rolling) for logical groups of components. There is an audit trail for all administration functions handled by the console including the control function, current status, start time, duration, initiator, and any error messages that resulted. In the future, you will also have the ability to roll-back to previous versions.

Operations can also be triggered by alerts. This scenario will be discussed in the Monitoring Module section.



The Administration Module allows you to perform control actions such as start, stop, and re-start on a single application or across an enterprise cluster.

MONITORING MODULE

The Monitoring Module is an optional component of the JBoss Network Enterprise Manager. It provides you an integrated view into the health of your JEMS infrastructure, JEMS-based applications, and other key IT dependencies including Apache Web Server and major operating systems. It also allows you to define alerts based upon pre-defined thresholds and, if so desired, automate control actions to correct faults. Graphing and charting including trend analysis along with a complete audit trail of alerts are also provided.

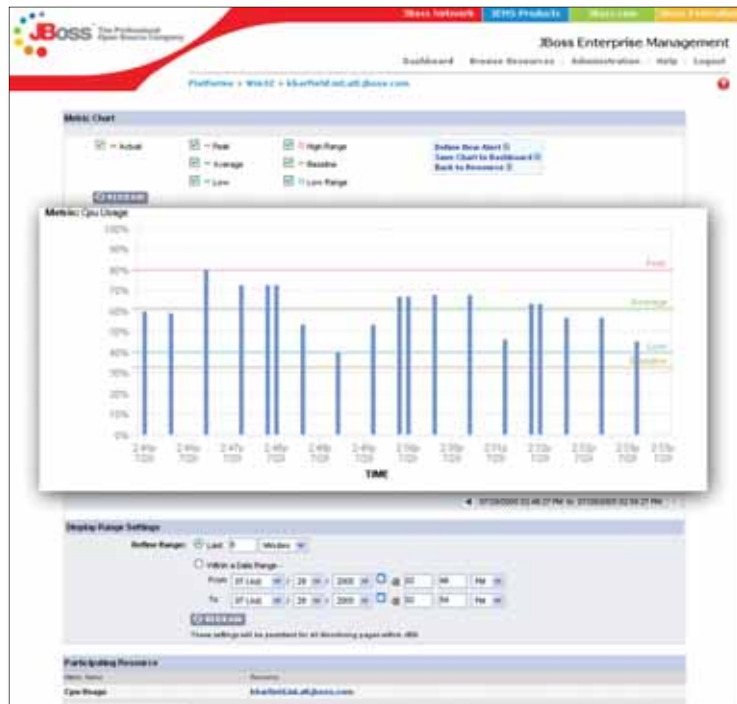
The Monitoring Module is continuously building and updating a model of behavior of every measurement collected by the Agents. Key pre-selected statistics are provided for each resource. These statistics are configurable and extendable. All data is stored in the repository. In addition to the core JEMS resources, the Monitoring Module can also monitor file systems and directories. Among other things, this allows you to monitor JBoss Configuration files looking for changes to files, file sizes, or permissions; ensuring a more secure JEMS environment.

The Monitoring Module GUI allows you to compare metrics from different resources across a common timeline. Metric charts provide a graphical representation of the metric data collected for each resource. Metrics can be charted using peak, low, average, and high value data. Additionally, charts can be used to demonstrate trends over time, which is useful for capacity planning, cost analysis, and other operational decisions.

Metrics are continually compared to baseline data allowing you to identify out of band problems. This allows you to quickly identify resources within your infrastructure that require attention as well as get a historical view of what was happening at the time of the fault. The Monitoring Module is responsible for firing events for inbound measurement reports, which are used by the alert engine to determine if previously defined alert criteria have been met.

The alerting engine allows you to create a wide range of conditional alert notifications. Alerts are currently delivered via SMTP / e-mail to defined users, groups, or to any generic e-mail address. Alerts via SNMP trap notification are expected to be released during Q4 2005. This will allow alerts from the JBoss Network Enterprise Manager to be shared with SNMP-compatible Network Management Systems such as BMC Patrol, CA Unicenter, HP OpenView, and IBM Tivoli.

Additionally, the Monitoring Module can kick off control actions based upon alerts. This is an advantage to having a fully integrated management platform. Alerts can both respond to control actions performed as well as perform control actions in response to alert conditions. Alerts can also be triggered on compound conditions and also set up to be dampened (or filtered) on a timed or number of occurrences basis. All alerts and corrective actions are logged and tracked within the JBoss Network Enterprise Manager repository; maintaining a comprehensive audit trail of alerts.



The Monitoring Module is a robust and extensible tool for monitoring your entire JEMS environment and related dependencies including operating systems, Apache Web Server, and Apache Tomcat.

UPDATE MODULE

The Update Module proactively notifies subscribers of new patches and software upgrades that are relevant to their specific environments. In many ways, the Update Module is similar to the Windows Update service applied to JEMS software. It allows you to deal with updates in a much more organized, efficient, and timely manner. Similar to syndicated update alerts from the JBoss Network Customer Portal, this Update Module eliminates the need for your organization to follow community threads, vendor portals, and the overwhelming influx of e-mail in order to learn about the latest updates.

Updates can be reviewed, rejected, or applied. They can be applied to individual components, instances, or clusters. They can also be applied to individual environments in a controlled fashion. This enables you to move first to QA, integration and test, and ultimately into production.

Software updates are a schedulable task within the Update Module. Updates can be applied in a rolling fashion or all at once. You can also roll-back to previous versions easily. Alerts can also be generated. For example, if an update to a particular component fails, you can be alerted.

The Update Module provides you version control over your JEMS environment and offers tremendous visibility over all components. It also provides an audit trail of every update to each component within your JEMS infrastructure.

The Update Module will be released as a beta in Q4 2005. Additional capabilities will be delivered in early 2006.

PROVISIONING MODULE

Planned for Q1 2006, the Provisioning Module will help you to configure new environments, compare and synchronize environments, and recreate existing environments quickly. The latter is especially useful for cases when you need to mirror your production environment in QA, a common requirement for many IT shops. This requirement is usually a challenge as is keeping environments synchronized.

In addition, the Provisioning Module will enable you to define workflow rules and processes for provisioning. You will be able to 1) configure new environments, 2) compare and synchronize environments, 3) recreate existing environments, and 4) roll back to previous states. A full audit trail will be maintained for each action. In addition, integrations to popular ticketing systems are planned.

More information on the Provisioning Module will be made available during the second half of 2005.

SUMMARY

The JBoss Network™ is a key component of the JBoss Subscription; a set of services and tools that assist you during every stage of your application lifecycle – from design and development through testing and deployment to on-going management and monitoring of their Middleware applications.

The JBoss Network Customer Portal acts as an online, self-service window into the JBoss support organization where you can more quickly obtain technical knowledge and download patches and updates. The JBoss Network Management Server is a comprehensive management and monitoring platform designed to dramatically improve the operational efficiency of your JEMS enterprise deployments. Together, the JBoss Network tools should dramatically simplify the amount of time and effort needed to manage your JEMS Middleware infrastructure and remove many of the associated business and technical risks.

Choose JBoss Network When You Need:

- » **Advanced integrated view** into the operations of your infrastructure.
- » **Quicker resolution to technical issues** by relying upon the Knowledge Module within the Customer Portal or by opening up a case with the industry's highest rated support team.
- » **Know what is installed where.** Inventory Module includes an auto-discovery mechanism which can peer into file system, process table and registries.
- » **Configure large environments.** From a single Administration Module console, you can apply administration operations to enterprise clusters.
- » **Visibility into the health of your applications.** Monitoring Module provides the visibility, collection, alerts, reporting and analysis of your enterprise environment.
- » **Keep current with software updates.** Update Module lets you know when there are updates available and provides the flexibility to manage the application of these updates.
- » **Add capacity on demand.** Provisioning Module provisions new instances of server types such as Apache, Tomcat and JBoss AS.
- » **Customized managed environment.** Built on an extendable architecture, JBoss Network Enterprise Manager is designed to make the addition of new monitored components simple.

Appendix I - JBoss Network Customer Portal Features

| Case Management Module Features | Registration Module Features |
|---|---|
| <ul style="list-style-type: none"> » Open, track, and close support cases » Instant online solution suggestions » Case history for auditing | <ul style="list-style-type: none"> » Register OS and JEMS components » Register versions » Subscribe to patch and update notifications » Subscribe to knowledge content alerts |
| Knowledge Module Features | Download Module Features |
| <ul style="list-style-type: none"> » Access to premium technical content » Access to certified technical solutions » Integrated search across JBoss technical resources » Integrated search across premium content » Integrated search across source code » Metadata-based rating system for search | <ul style="list-style-type: none"> » Download latest production-ready JEMS software » Download certified binary patches » Download upgrades » Download notifications » Audit trail (Q4 2005) |

Appendix II - JBoss Network Enterprise Manager Features

| Management Console Features | Inventory Module Features | Administration Module Features |
|---|--|---|
| <ul style="list-style-type: none"> » User/group control » Custom dashboards » Portal or command line interface » System/alert configuration » Graphical view » Extensive command line interface » Auditing » Integrated map » Search for resources » Online and static help » Built-in database / data repository | <ul style="list-style-type: none"> » Catalog JEMS assets » Auto-discovery » Discover by File Scan » Discover by Registry Scan » Discover by Process Table Scan » Schedulable discovery » Manual management | <ul style="list-style-type: none"> » JEMS control operations (start, stop, re-start) » Apply across single application or cluster » Define order of control actions » Group-based operations (parallel, serial/ordered) » Schedulable operations (on-demand, recurring) » Operations triggered by alerts » Configuration and application repository » Roll-back |
| Monitoring Module Features | Software Update Features | Provisioning Features |
| <ul style="list-style-type: none"> » Pre-selected statistics » Configurable statistics » Extensible/custom statistics » Alert definition » Dampening (max ceiling of alerts) » Control actions on alerts » Baseline measurements » Baseline problem detection » Graphing and charting » Trend analysis » Audit trail on alerts » OS support (all major OS) » JEMS support » Apache support » Plug-in Developer's Kit » Nagios/scripting plug-in for monitoring » JBoss Network port monitoring | <ul style="list-style-type: none"> » Apply updates to individual components or instances » Apply updates to clusters » Apply updates to selected environments (e.g. QA) » Apply updates in a rolling fashion » Apply updates all at once » Schedulable deployments » Pre/Post scripting » Alerting » Version control » Roll-back » Maintain audit trail » Access to JBoss repository » Access to local repository | <ul style="list-style-type: none"> » Replication of new JEMS environments » Create new environments » Compare and synchronize environments » Cluster actions » Define workflow rules, process definition » Maintain audit trails » Integrate into ticketing systems » Roll-back » JBoss support » Tomcat Support |

Appendix III - JBoss Network Enterprise Manager – System Requirements

| | Small Deployment | Medium to Large Deployment |
|---------------------------|--|---|
| Operating System | Linux, Solaris, HP/UX, Windows | Linux, Solaris, HP/UX, Windows |
| Processor | 1x 1GHz x86 or 1 x 450MHz sparc or 1 x 440MHz PA-8800 | 2x 2GHz x86 or 2 x 450MHz sparc or 2 x 440MHz PA-8800 |
| Memory | 512 MB | 1 GB |
| Storage | 10 GB | 40 GB |
| Supported Agent Platforms | Windows, Linux, AMD-Linux, Solaris, HP/UX, AIX | |
| Supported Servers | JBoss AS 3.2x, 4.x; Tomcat 4.0, 4.1, 5.0; Apache 1.3, 2.x; IIS 4.x, 5.x, 6.x; iPlanet 4.1, 6.0 | |
| Required JDK | JDK required is 1.4.x for all deployments | |
| Linux Supported Versions | RedHat Enterprise 2.1/3.0, RedHat Linux 6.2 - 9.0, SuSe Enterprise 8.1, Fedora Core 2, Core 3, Gentoo Linux | |
| Supported Databases | Postgres Version 8.0 or above (Ships with the JBNetwork Enterprise Manager). Oracle 8, 9i | |

Appendix IV - Monitoring Module Detailed Feature List

| | |
|--|--|
| <p>Supported Systems</p> <ul style="list-style-type: none"> » JBoss AS (3.2.X, 4.X) » Apache Tomcat (4.0, 4.1, 5.0) » Apache Web Server (1.3, 2.0, ERS 2.3, ERS 2.4) » Novell / SuSe Enterprise 8.1 » Red Hat Linux 6.2 - 9.0 » Fedora Core 2, Core 3 » Gentoo Linux » Windows NT 4.0 » Windows 2000 » Windows XP » Windows 2003 Server / Advanced Server / Terminal Server » Sparc Solaris 2.4 – 10 » X86 Solaris 2.4 – 10 » HP/UX 11.X » IBM AIX (4.3, 5.1, 5.2) » Nagios/scripting plug-in | <p>Alerting</p> <ul style="list-style-type: none"> » Driven from metrics or administration actions » Condition type <ul style="list-style-type: none"> • Greater, less, equal • Compared against baselin • Value changes » Multiple conditions supported (and/or) » Message dampening to reduce alert overload <ul style="list-style-type: none"> • Recovery alerts <ul style="list-style-type: none"> * Each time conditions are exceeded or met * When alerts exceed limit over defined time period » Action filter <ul style="list-style-type: none"> • Disable alert until re-enabled manually or by recovery alert • Disregard control actions that are defined for related alerts • Filter notification actions that are defined for related alerts » Notification to groups, users and SMTP » SNMP trap notification (Q3 2005) |
| <p>Metric Types</p> <ul style="list-style-type: none"> » Availability - ready to service requests » Usage - resource usage counters » Performance - response time » Utilization - rates of usage | <p>Baselines</p> <ul style="list-style-type: none"> » Tracking normal resource behavior » Dynamic rolling baselines » Static overrides |
| <p>Metric Portal View</p> <ul style="list-style-type: none"> » Current health portlet » Child resources health portlet » Host resource health portlet » Problem metrics portlet » Group metric views | <p>Metric Charting</p> <ul style="list-style-type: none"> » Metric range specification » Single metric single resource » Multiple metrics single resource » Single metric multiple resources » Chart Legend includes - actual, peak, average, low, low range, high range and baseline |
| <p>Problem Resources</p> <ul style="list-style-type: none"> » Tracking resources that fall outside the baseline range » Alert based on variation of baseline values | <p>Miscellaneous</p> <ul style="list-style-type: none"> » Audit trail » Agent plug-in developer's kit » JBoss Network Customer Portal port monitoring |

Appendix V - JBoss Application Server Metrics

| General Server Matrix | JCA Connection Pool Metrics |
|---|---|
| <ul style="list-style-type: none"> » Active Thread Count » Active Thread Group Count » JVM Free Memory » JVM Total Memory » JVM Max Memory » JMS Message Cache Size » JMS Message Cache Hits » JMS Message Cache Hits per Minute » JMS Message Cache Misses » JMS Message Cache Misses per Minute » Availability » Transactions Active » Transactions Committed » Transactions Committed per Minute » Transactions Rolled back » Transactions Rolled back per Minute » JMS Message Cache Current Memory Usage » JMS Message Cache High Memory Mark » JMS Message Cache Max Memory Mark | <ul style="list-style-type: none"> » Min Connections » Max Connections » Total Connections » Available Connections » Connections Created » Connections Destroyed » Availability |
| | Stateful Session EJB Metrics |
| | <ul style="list-style-type: none"> » Create Calls » Remove Calls » Method-Ready Beans » Passive Beans » Availability |
| Entity EJB Metrics | JMS Destination Metrics |
| <ul style="list-style-type: none"> » Create Calls » Create Calls per Minute » Remove Calls » Remove Calls per Minute » Ready Beans » Pooled Beans » CacheSize » PassivatedCount » PassivatedCount per Minute » PoolSize » MaxPoolSize » Availability | <ul style="list-style-type: none"> » Messages in Queue » Availability » Receivers Count |
| | Message Driven EJB Metrics |
| | <ul style="list-style-type: none"> » Create Calls » Create Calls per Minute » Remove Calls » Remove Calls per Minute » Messages Received » Messages Received per Minute » Availability |

Appendix VI - Tomcat Monitoring Metrics

| Reliability Metrics | Resource Utilization Metrics |
|---|--|
| <ul style="list-style-type: none"> » Uptime » Availability | <ul style="list-style-type: none"> » JVM Active Thread Count » JVM Active Thread Group Count » JVM Free Memory » JVM Total Memory » Process CPU System Time » Process CPU User Time » Process Memory Size » Process Shared Memory Size » Process Open File Descriptors |
| General Server Metrics | |
| <ul style="list-style-type: none"> » Number of Requests Served » Number of Requests Served per Minute » Total Processing Time » Total Processing Time per Minute | |
| Tomcat 4.1 and 5.0 Connector Metrics | Tomcat 4.0 Webapp Metrics |
| <ul style="list-style-type: none"> » Availability » Bytes Received » Bytes Received per Minute » Bytes Sent » Bytes Sent per Minute » Error Count » Error Count per Minute » Request Count » Request Count per Minute » Maximum Request Time » Processing Request Time » Processing Request Time per Minute » Threads Allocated » Threads Active | <ul style="list-style-type: none"> » Number of Requests Served » Number of Requests Served per Minute » Number of errors » Number of errors per Minute » Sessions Created » Sessions Created per Minute » Sessions Destroyed » Sessions Destroyed per Minute » Minimum Response Time of a Servlet » Maximum Response Time of a Servlet » Average Response Time » Total Processing Time for the webapp » Total Processing Time for the webapp per Minute » Availability |
| Tomcat 4.1, 5.0 and 5.5 Webapp Metrics | Tomcat 4.0, 4.1, 5.0 and 5.5 Servlet Metrics |
| <ul style="list-style-type: none"> » Number of Requests Served » Number of Requests Served per Minute » Number of errors » Number of errors per Minute » Sessions Created » Sessions Created per Minute » Sessions Destroyed » Sessions Destroyed per Minute » Minimum Response Time of a Servlet » Maximum Response Time of a Servlet » Average Response Time » Total Processing Time for the webapp » Total Processing Time for the webapp per Minute » Availability » Expired Sessions » Expired Sessions per Minute » Rejected Sessions » Rejected Sessions per Minute » Max Active Sessions | <ul style="list-style-type: none"> » Availability » Request Count » Request Count per Minute » Error Count » Error Count per Minute » Maximum Response Time » Average Response Time » Min Response Time » Processing Request Time » Processing Request Time per Minute |

Copyright

Copyright © 2005 JBoss, Inc. All Rights Reserved.

RESTRICTED RIGHTS NOTICE

This document may not, in whole or in part, be reproduced, photocopied or translated without the expressed written consent of JBoss, Inc. This document is subject to change without notice.

TRADEMARKS

JBoss and JEMS are registered trademarks or trademarks of JBoss, Inc. in the United States and other countries. All other trademarks or registered trademarks herein are property of their respective owners.

