

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[¡NUEVO!] 2008-04-01



2008-03-25



2008-03-17



2008-03-11

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Tutorial desarrollado por



Iván García Puebla

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero Técnico en Informática de Gestión por la Universidad de Valladolid.

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

Descargar este documento en formato PDF: [monitorizarJBossJMX.pdf](#)

Fecha de creación del tutorial: 2008-04-05

La API JMX y Monitorización de JBoss

En este tutorial vamos a hacer una breve introducción a JMX y de cómo podemos acceder a las características de JBoss gracias a los MBeans que implementan esta API. Asimismo presentamos tres herramientas de monitorización del servidor.

Si estás interesado en monitorizar sólo Tomcat, visita este tutorial de nuestro compañero [Javier Antoniucci: Monitorización de Tomcat con JMX](#)

Contenidos:

1. [La API JMX y Monitorización de JBoss](#)
 1. Contenidos:
 2. Introducción a JMX
 1. Arquitectura de JMX
 3. Monitorización de JBoss
 1. Monitorización con JConsole
 2. Monitorización con MC4j Management Console for Java
 3. Monitorización con ManageEngine Applications Manager 8
 4. Conclusiones

Introducción a JMX

La **Java Management eXtensions (JMX)** API es un estándar de Sun para la gestión y monitorización remota de recursos java como:

- Aplicaciones y dispositivos
- Servidores y servicios
- JVM

que nos permite entre otras cosas consultar o cambiar una determinada configuración, conocer estadísticas y comportamiento de una aplicación, conocer cambios de estado (y notificarlo, iniciar acciones, etc), crear nuestros manejadores de recursos y publicarlos en una API, interoperar con otras tecnologías...

JMX se integró pronto en la J2EE y más tarde en la JSE, concretamente desde la JDK 5. Su jerarquía de subpaquetes se encuentran bajo `javax.management`, y esta es su API pública. Asimismo existe una API privada bajo `com.sun.jmx`, a la que sólo debería tener acceso las propias librerías de la JDK, puesto que esta implementación puede evolucionar y ser cambiada en futuras versiones.

A partir de la JDK 5.0 se incluye `Jconsole` para monitorizar la JVM y aplicaciones que implementen JMX. Es un ejecutable con interfaz gráfica que se encuentra en la carpeta `bin` de la instalación de la distribución de Sun JDK.

Arquitectura de JMX

Catálogo de servicios Autentia (PDF 6,2MB)



[En formato comic...](#)



☐ Web
☒ www.adictosaltrabajo.com

Últimos tutoriales

2008-04-05
[JMX y monitorización de JBoss](#)

2008-04-05
[Jersey: la implementación de RESTful de Sun](#)

2008-04-05
[Metro: pila de webservices de Sun. Integración con Maven 2](#)

2008-04-05
[Metro: pila de webservices de Sun.](#)

2008-04-04
[Espectaculares efectos visuales en el escritorio de Linux, con Compiz Fusion](#)

2008-04-04
[Monitorización de Web Services con Glassfish Wsmonitor](#)

2008-04-04
[Axis2. Ejemplo de creación de un servicio Web](#)

2008-04-03
[Servicios Web RESTful en Axis 2](#)

2008-04-03
[XML Signature - Firma Digital sobre XML](#)

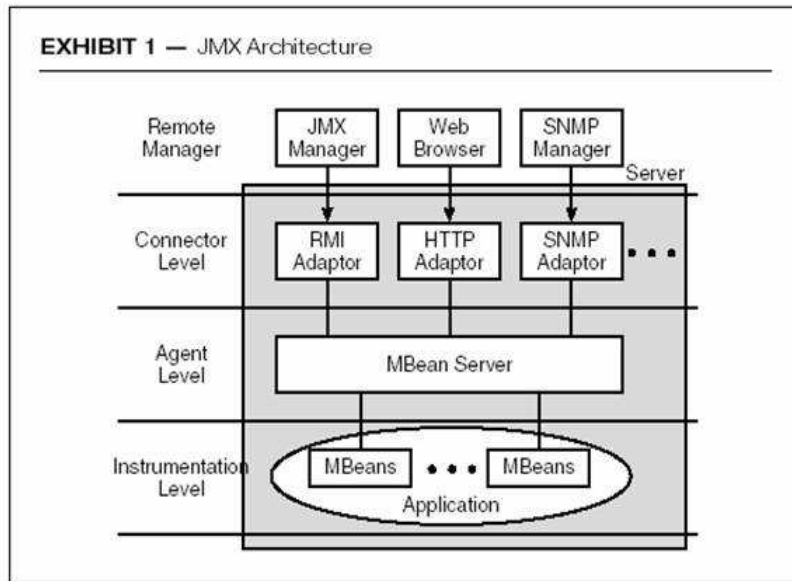
2008-04-03
[XML Encryption, Criptografía sobre XML](#)

Últimas ofertas de empleo

2008-04-04
[Banca - Genérico - MADRID.](#)

Arquitectura basada en 3 capas:

1. **Capa de aplicación (o nivel de instrumentación)**, la capa de más bajo nivel en la que residen los componentes (*MBean*) que facilitan la información necesaria para la gestión de una aplicación. Estos componentes son desarrollados según las necesidades de gestión específicas de cada parte de una aplicación.
2. **Nivel de agente**: facilita una interfaz para el manejo de los *MBean* del nivel de instrumentación.
3. **Nivel de Adaptadores**: uno o más conectores (o adaptadores de protocolo) que proporcionan acceso desde los sistemas de monitorización remotos.



2008-04-03
[Banca - Genérico - MADRID.](#)

2008-04-02
[T. Información - Analista / Programador - MADRID.](#)

2008-04-02
[T. Información - Analista / Programador - MADRID.](#)

2008-03-29
[T. Información - Analista / Programador - MADRID.](#)

Anuncios Google

Monitorización de JBoss

JBoss implementa JMX en su microkernel. El propio servidor dispone de una consola JMX accesible por la URL <http://localhost:8080/jmx-console> que muestra los *MBeans* públicos accesibles por ella o programáticamente por código Java.:



JMX Agent View PC-TRABAJO

ObjectName Filter (e.g. "jboss:*", "*:service=invoker,*") : ApplyFilter

Catalina

- ♦ [type=Server](#)
- ♦ [type=StringCache](#)

JMImplementation

- ♦ [name=Default,service=LoaderRepository](#)
- ♦ [type=MBeanRegistry](#)
- ♦ [type=MBeanServerDelegate](#)

com.arjuna.ats.properties

- ♦ [module=arjuna](#)
- ♦ [module=jta](#)
- ♦ [module=txoj](#)

jboss

- ♦ [database=localDB,service=Hypersonic](#)
- ♦ [name=PropertyEditorManager,type=Service](#)
- ♦ [name=SystemProperties,type=Service](#)
- ♦ [readonly=true,service=invoker,target=Naming,type=http](#)
- ♦ [service=AttributePersistenceService](#)
- ♦ [service=ClientUserTransaction](#)
- ♦ [service=JNDIView](#)
- ♦ [service=KeyGeneratorFactory,type=HiLo](#)
- ♦ [service=KeyGeneratorFactory,type=UUID](#)

Algunas acciones útiles:

- Mostrar el árbol JNDI
- Forzar un volcado de memoria
- Mostrar el uso del pool de memoria
- Gestionar el escáner de despliegues
- Redespargar una aplicación

Anuncios Google

[Tutorial Unix](#)

[Tutorial Dreamweaver](#)

[Manual HTML](#)

[Manual Programación](#)

[Tutorial HTML](#)

- Acceder a la base de datos Hypersonic
- Detener JBoss
- Conocer estado de los EJB desplegados e instanciados
- ... y mucho más

El JBoss Web Console (<http://localhost:8080/web-console/>) muestra también la consola JMX en jerarquía de árbol:

The screenshot shows the JBoss Management Console interface. On the left is a tree view of JMX MBeans, including categories like Monitoring, J2EE Domains, AOP, System, Unified ClassLoaders, and JMX MBeans. Under JMX MBeans, there are sub-items like Catalina, JMIImplementation, com.arjuna.ats.properties, jboss, jboss.admin, jboss.alerts, jboss.aop, jboss.bean, jboss.beans, jboss.cache, jboss.console, jboss.deployer, and jboss.deployment. The main panel on the right displays the 'JBoss™ Application Server' status. It includes a 'JBoss' section with version (4.2.2.GA), environment (start date, host, base location), and JVM - Hardware section with hardware (CPU, OS) and JVM environment (free/max memory) details.

Monitorización con JConsole

Jconsole (www.servletsuite.com) usa el JMX de JBoss para monitorizar el servidor: estado, logs, despliegues, uploads y visor de los MBeans:

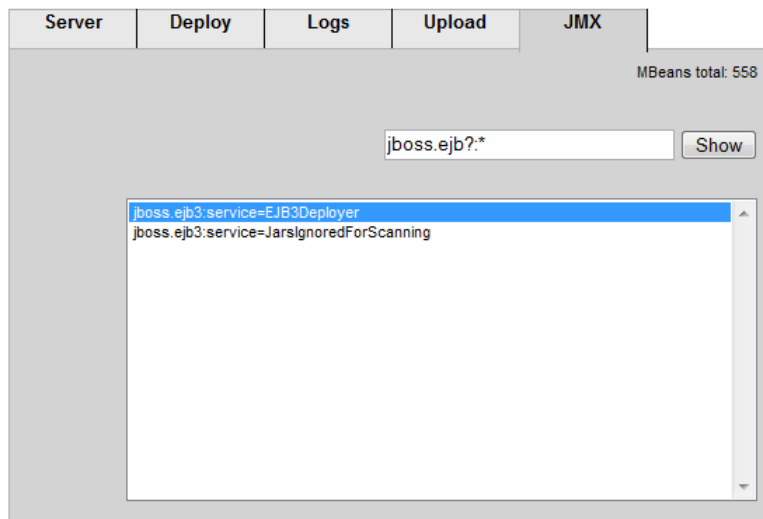
The screenshot shows the JConsole application window. At the top are tabs for Server, Deploy, Logs, Upload, and JMX. The 'Server' tab is active, showing 'Uptime: 0d 3:54' and a 'Shutdown' button. Below this is a 'Memory' section with a vertical bar chart showing 53% used (green) and 47% free (red). To the right is an 'Info' box containing system details: Version (4.2.2.GA), SVNTag (JBoss_4_2_2_GA), date (200710221140), Host (PC-TRABAJO), JDK (1.5.0_14), OS (Windows Vista), Total memory (154861568), and Threads (64). At the bottom, it says '© Coldbeans ver. 1.7 License: [license.htm](#) Updates: www.servletsuite.com'.

Para poder trabajar con JConsole debemos seguir los siguientes pasos:

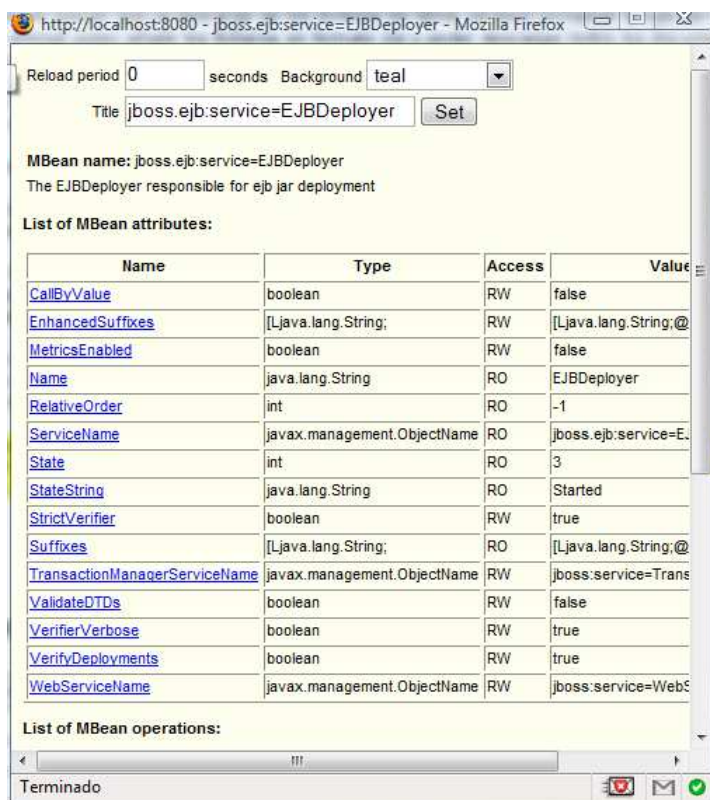
1. Descargar jconsole.war desde <http://www.servletsuite.com/jmx/jconsole.htm>
2. Copiar jconsole.war en la carpeta deploy del servidor utilizado, para su despliegue (habitualmente en /jboss-4.2.2.GA/server/default/deploy/)
3. Con JBoss en ejecución, acceder a <http://localhost:8080/jconsole>

Cuando accedamos a la pestaña 'JMX' se nos presentará una casilla para introducir un patrón que determinará los MBeans que visualizaremos en la lista inferior. Los patrones son los habituales:

- Nombre totalmente cualificado e.g: boss.j2ee:service=EARDeployer
- Wildcard: *, ? e.g.: jboss.jm?:*, jboss:*, jboss.*:*, *:*



Simplemente hay que hacer click sobre un MBean obtenido en la lista para acceder a sus detalles:



Monitorización con MC4j Management Console for Java

El **MC4j Management Console for Java** (open source) es una aplicación standalone basada en java diseñada para monitorizar múltiples servidores y visualizar sus los MBeans, en su caso.



Para instalar la última versión (en el momento de escribir este tutorial, 2.0 alpha 1) sobre JBoss 4.2, tenemos que hacer un cambio en la librería log4j debido a una incompatibilidad de versiones.

1. Descargar la versión apropiada a nuestro sistema operativo desde <http://www.mc4j.org/confluence/display/MC4J/Download>

- y descomprimir
- 2. Descargar la librería log4j de <http://logging.apache.org/log4j/1.2/>, descomprimir y localizar el log4j-1.2.*.jar
- 3. Renombrar a log4j-1.2.8.jar
- 4. Sustituir el existente en mc4j/mc4j/modules/ext por el renombrado anterior

Finalmente cargamos la aplicación con el ejecutable situado en mc4j/bin

como hemos dicho antes MC4J soporta varios servidores, y tenemos que configurar una conexión de monitorización a cada uno de ellos que nos interese. En este caso para nuestro JBoss (en ejecución), pulsamos primero sobre el botón de añadir conexión al servidor:



En la siguiente pantalla del asistente seleccionamos el tipo de servidor y damos un nombre a la nueva configuración que estamos creando. Con una instalación estándar de JBoss no es necesario cambiar ninguno de los demás parámetros aparecen en las siguientes casillas de JNDI, contexto, etc.:

Panel Name wizard (1 of 3)

JBoss

Start by selecting your server connection type above

Name: Monitor de JBoss

JNDI Name: jmx/rmi/RMIAdaptor

Initial Context Factory: org.jnp.interfaces.NamingContextFactory

Server URL: jnp://localhost:1099

Principle:

Credentials:

Advanced...

En el paso 2 del asistente seleccionamos la carpeta raíz del JBoss que queramos monitorizar:

Please select a server installation for JBoss. If there are none available, press "Add" to the right and browse to the correct installation directory.

JBoss - C:\Tutoriales\jboss-4.2.2.GA

Add

Remove

El último paso sirve para crear el listado de dependencias de Servidor. Si hemos seleccionado correctamente la instalación del servidor en el paso 2, MC4J las encontrará por nosotros y no tendremos que hacer nada:

Panel Name wizard (3 of 3)

Custom classpath and server libraries

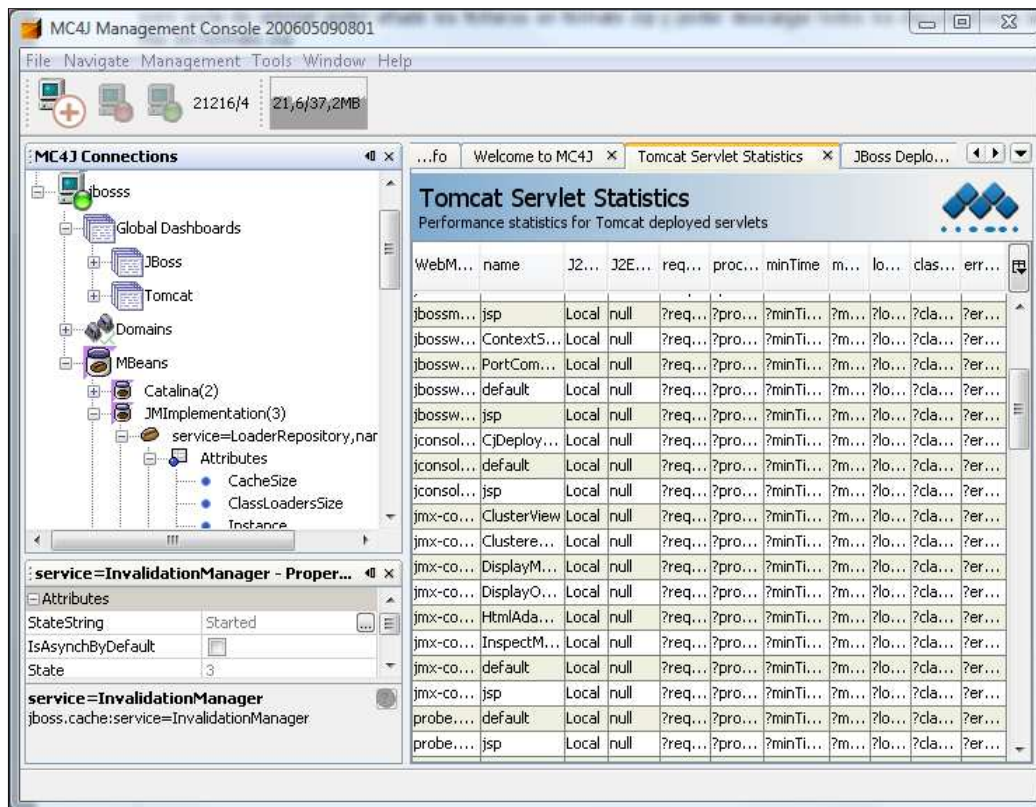
- C:\Tutoriales\jboss-4.2.2.GA\lib\jboss-jmx.jar
- C:\Tutoriales\jboss-4.2.2.GA\lib\jboss-system.jar
- C:\Tutoriales\jboss-4.2.2.GA\client\jbossall-client.jar
- C:\Tutoriales\jboss-4.2.2.GA\client\log4j.jar
- C:\Tutoriales\jboss-4.2.2.GA\client\concurrent.jar
- C:\Tutoriales\jboss-4.2.2.GA\client\jboss-jsr77-client.jar
- C:\Tutoriales\jboss-4.2.2.GA\client\jnp-client.jar

Additional mc4jlib directory libraries

Add...

Remove

Hemos llegado al final. La configuración aparecerá en el árbol izquierdo de la interfaz, y si le damos al botón verde estableceremos la conexión, como se aprecia en la imagen siguiente:



Monitorización con ManageEngine Applications Manager 8

ManageEngine Applications Manager es un potente monitor comercial con una versión libre (limitación: máximo 5 monitores) aplicable a servidores, BBDD, aplicaciones, websites, ERP, transacciones y mucho más. Ha sido diseñada con una interfaz muy completa y ofrece múltiples estadísticas, que pueden asociarse acciones al seguimiento de recursos, como alarmas y notificaciones por correo electrónico. Permite la monitorización de recursos remotos (los anteriores monitores se limitaban al servidor local) y la monitorización de MBeans de JMX, por supuesto :-)

iComencemos con esta interesante herramienta!

- Descarga: http://manageengine.adventnet.com/products/applications_manager/download.html y ejecutar instalable
- Instalación (para ManageEngine AM 8 y Jboss 4.2, puertos estándar):
 - Descargar log4j de <http://logging.apache.org/log4j/1.2/>, descomprimir y localizar el log4j-1.2.*.jar
 - Renombrar a log4j.jar
 - Sustituir el existente en AdventNet/ME/AppManager8/working/classes/ por el renombrado anterior
- Ejecución: startApplicationsManager en AdventNet/ME/AppManager8
- Jboss ha de poder aceptar conexiones remotas (e.g. iniciar el servidor con `run -b 0.0.0.0` o la IP que deseemos que tenga acceso)

ManageEngine se habrá levantado en nuestro sistema como un servidor de monitorización, por lo tanto su consola de configuración será accesible vía web. Por eso navegamos a la URL <http://localhost:9090/> y en la pantalla de navegación nos autenticamos con login admin y password admin.

Vamos a configurar ahora un monitor para nuestro JBoss. Primero pulsamos en la pestaña Monitors | New Monitor y en la columna Application Servers pulsamos sobre JBoss:



Seleccionamos como tipo de monitor JBoss Server e introducimos los datos como se observa en la imagen. (NOTA: escogiendo la versión 4.X la aplicación nos obliga a copiar Jboss-4.2.2.GA/client/jbossall-client.jar en AdventNet/ME/AppManager8/working/classes/jboss/40/, si este directorio no existe hay que crearlo):

Add Monitor of type JBoss Server

Host Name / IP Address*

SubNet Mask* [Advanced ▾](#)

Port*

SSL is enabled ☐

JBoss Version* ▾

Polling Interval* Minute(s)

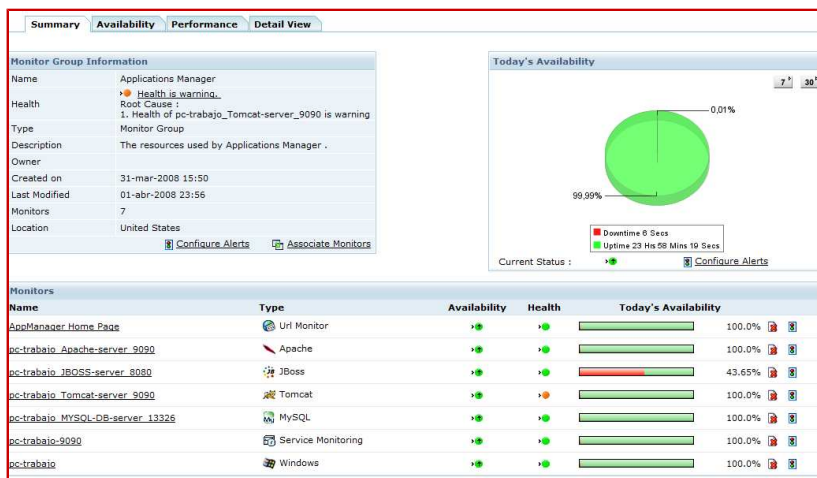
Authentication is enabled ☐

Associate Monitor Instance to Monitor Group

Select the Monitor Group [Create New Monitor Group](#)

Add Monitor(s) **Restore Defaults**

Una vez conectado en nuevo monitor a JBoss, se muestra un resumen de monitores (pulsas sobre la imagen para verla en su tamaño original):



Si pulsamos sobre el monitor de JBoss veremos el siguiente panel de resumen. En él pueden asociarse alarmas a los recursos mostrados, añadir atributos y MBeans a mostrar... se trata de un panel muy configurable.





Conclusiones

Como cierre a este tutorial podemos quedarnos con estos puntos fundamentales:

- La API JMX permite acceder a información publicada por una aplicación
- Las unidades que publican esa información son los Mbeans
- JBoss implementa multitud de MBeans
- Existen herramientas para usar la JMX de JBoss y monitorizar su estado

En **Autentia** ofrecemos permanentemente formación de las últimas tecnologías del sector informático. Si quieres **formación adaptada a tus necesidades y desarrollos**, no dudes en visitar [nuestros cursos](#) y contactar con nosotros; ¡trabajamos con estas tecnologías en nuestro día a día!

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- Añadir a favoritos Technorati. [ADD THIS BLOG TO MY FAVORITES](#)



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com



Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de suscripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Manejo de JMS en JBOSS	Este tutorial nos dará un poco más de luz sobre JMS, un sistema de mensajería de SUN, sobre un servidor JBOSS con algunos ejemplos muy explicativos	2007-03-05	3682	pdf
Guía rápida de instalación de JBOSS Application Server 4.	En este manual veremos paso a paso la forma de instalar en tu equipo JBoss Application Server 4.	2006-11-02	4238	pdf
MBeans y JBoss	Este tutorial tiene como finalidad familiarizarse con el servidor de aplicaciones JBoss y su API JMX	2007-01-17	3454	pdf
Planificar tareas en JBoss	En este tutorial os enseñaremos a planificar tareas periódicas con JBoss	2006-06-07	5120	pdf
XMBeans y JBoss	Este tutorial es una continuación de una anterior llamado MBeans y JBoss	2007-01-18	2644	pdf
Slimming básico de JBoss	En este tutorial se muestra un ejemplo concreto y sencillo de mejora en el tiempo de arranque de JBoss	2008-02-07	687	pdf
Monitorización de Tomcat con JMX	Este tutorial nos cuenta, paso a paso, como monitorizar Tomcat usando su implementación JMX	2007-01-29	2366	pdf
Instalar JBoss	Os mostramos como instalar en servidor gratuito de aplicaciones JBOSS así como a automatizar su arranque y parada.	2003-06-30	25801	pdf
Instrumentación de componentes Java usando JMX	Os mostramos como utilizar las características avanzadas de los servidores de aplicaciones Java/J2EE para realizar la instrumentación de componentes Java usando JMX y poniendo como ejemplo la gestión dinámica de propiedades entre nodos	2005-11-21	10219	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.