

¿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)

	Hosting Patrocinado por enREDados.com 
---	--

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

Tutorial desarrollado por: spbravo@gmail.com	 real business solutions
---	---

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

Módulos de automatización
Periferia y comunicación de PLCs
en el entorno Siemens.

Jmx Management Console
JMX Consoles & SNMP Consoles
JBossMX, WebLogic JMX, MX4J,
JMX1.2

Servidor Virtual Dedicado
Soluciones Intel y VMware para
infraestructuras virtuales

Download Java Report Tool
Drag & drop Report creation for
Java applications. Powerful charts!

Anuncios Goooooogle

Anunciarse en este sitio

INSTRUMENTACIÓN DE COMPONENTES JAVA USANDO JMX

ÍNDICE

<i>Capítulo/Sección</i>	<i>Página</i>
1. INTRODUCCIÓN	
2. LA SOLUCIÓN JMX	
2.1 INSTRUMENTACIÓN DISTRIBUIDA BASADA EN NOTIFICACIONES JMX	
2.2 IMPLEMENTACIÓN	
2.3 CONCLUSIONES	
-	

1. INTRODUCCIÓN

Una de las necesidades que se tienen en cualquier aplicación es la de poder cambiar el comportamiento de alguno de los componentes en caliente. ¿Quién no ha necesitado activar o desactivar el servicio de auditoría, cambiar el nivel de traza o algún parámetro de configuración?. Seguramente todo ingeniero o arquitecto software encontrará estos problemas como algo muy común en su trabajo. También es seguro que existen diferentes soluciones cada una con sus pros y contras.

En este pequeño informe vamos a intentar dar un enfoque un poco diferente a los usados tradicionalmente para gestionar los servicios de las aplicaciones. Nos vamos a basar en la tecnología JMX y en las capacidades que nos dan los servidores de aplicaciones comerciales.

Aunque la solución propuesta podría tener más usos, nos vamos a centrar en la problemática de realizar los cambios en caliente en un entorno de alta disponibilidad en el que se dispone de varios servidores en cluster.

Para centrar el escenario en el que nos vamos a mover, debemos tener en cuenta que los servidores de aplicaciones más importantes del mercado usan la tecnología JMX para manejar sus propios procesos internos. En este informe pondremos como ejemplo el servidor Websphere en su versión 5.0 pero los conceptos mostrados aquí podrían ser trasladables a cualquier otro servidor que diese soporte a la tecnología JMX .

2. LA SOLUCIÓN JMX

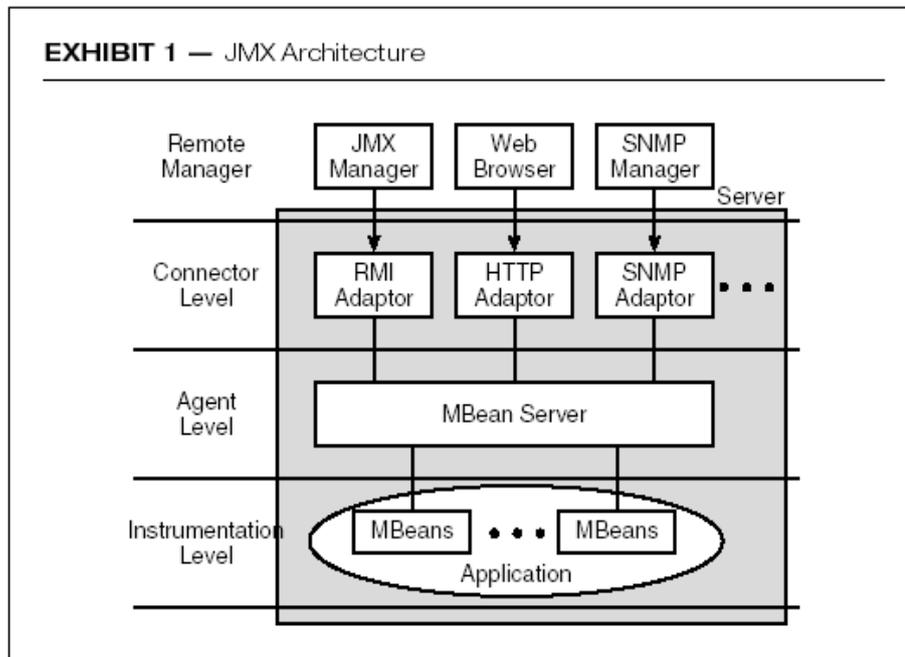
Como requisito previo a implantar la solución de cambio en caliente de propiedades en un entorno distribuido, se necesitaría que los componentes que desempeñan servicios susceptibles de ser instrumentados se codificasen como MBeans o al menos publicasen su interfaz en un MBean de tal manera que la invocación de un método del MBean actualizase el estado del componente.

¿Pero qué es JMX?

Java Management Extensions (JMX) es la especificación java que define la arquitectura de operación, que facilita la gestión de aplicaciones y servicios. Esta tecnología permite que los desarrolladores Java integren sus aplicaciones con las soluciones existentes de gestión y operación como veremos más adelante.

¿Cuál es la arquitectura de JMX?

La arquitectura JMX define tres niveles:



- Nivel de Instrumentación (Capa de Aplicación)
 - En la capa de aplicación (llamado *nivel de instrumentación*), es donde residen los componentes que facilitan la información necesaria para la gestión de una aplicación. Estos componentes son desarrollados según las necesidades de gestión específicas de cada aplicación. Por ejemplo, un componente puede tener un método para parar un servicio dentro de una aplicación.
- Nivel de Agente
 - El *nivel de agente* es el que facilita una interfaz para el manejo de los componentes del nivel de instrumentación.
- Nivel de Adaptadores
 - El *nivel de adaptadores*, consiste en uno o más conectores (o adaptadores de protocolo) que proporcionan acceso desde los sistemas de monitorización remotos

Pero, ¿qué es un MBean?

Los Managed Beans o MBeans son los objetos JMX que exponen la información manejable en forma de propiedades y operaciones.

Hay varios tipos de MBeans (Estandar MBean, Dynamic MBean y Model MBean) y deberemos escoger uno u otro según las necesidades de instrumentación que tengamos.

Pero yo ya tengo servicios desarrollados. ¿Puedo instrumentarlos con JMX?

Para mostrar con código Java como podríamos instrumentar un servicio ya existente, os pongo el siguiente ejemplo en donde se instrumenta nada menos que una HashMap:

```
MBeanServer mbs = MBeanServerFactory.createMBeanServer();
// The MBean Server

HashMap map = new HashMap();
// The resource that will be managed

// Construct the management interface for the Model MBean
Method getMethod = HashMap.class.getMethod("get", new Class[] {Object.class});
ModelMBeanOperationInfo getInfo =
    new ModelMBeanOperationInfo("Get value for key", getMethod);
ModelMBeanInfo mmbi =
    new ModelMBeanInfoSupport(HashMap.class.getName(),
        "Map of keys and values",
        null, // no attributes
        null, // no constructors
        new ModelMBeanOperationInfo[] {getInfo},
        null); // no notifications

// Make the Model MBean and link it to the resource
```

```

ModelMBean mmb = new RequiredModelMBean(mmbi);
mmb.setManagedResource(map, "ObjectReference");
// Register the Model MBean in the MBean Server
ObjectName mapName = new ObjectName(":type=Map,name=whatever");
mbs.registerMBean(mmb, mapName);

```

La parte interesante es la que está en rojo.

```
mmb.setManagedResource(map, "ObjectReference");
```

Como se puede ver, partiendo de un objeto ya existente (la HashMap) se le dice al MBean que la instrumente.

Una forma de realizar esta instrumentación de manera sencilla usando Model-MBeans se puede consultar en el proyecto <http://jakarta.apache.org/commons/modeler/>.

2.1 INSTRUMENTACIÓN DISTRIBUIDA BASADA EN NOTIFICACIONES JMX

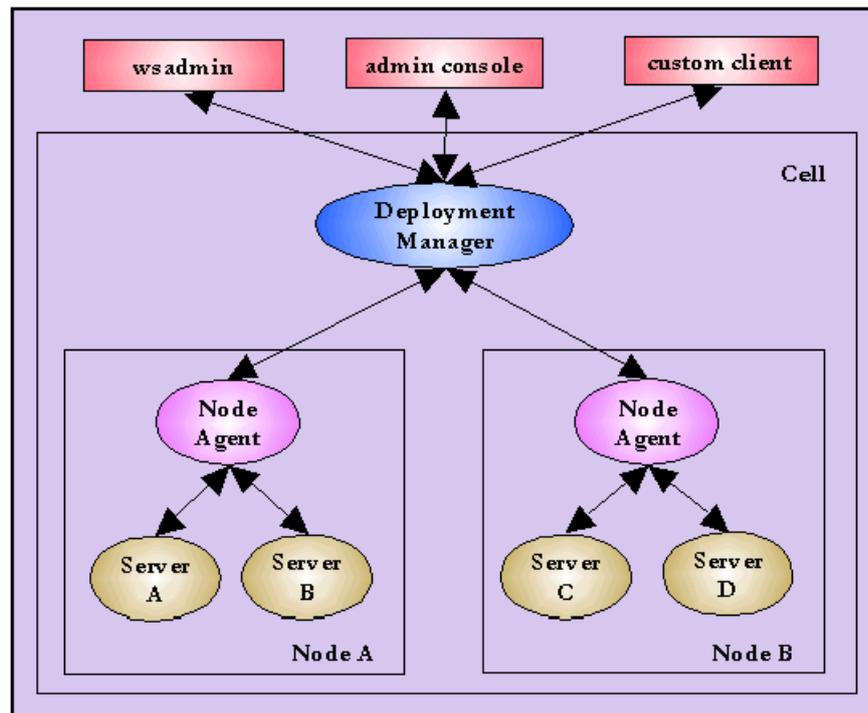
Una vez que se ha mostrado que los MBeans permiten instrumentar servicios, el principal reto que plantea un entorno cluster es la propagación de los cambios que sufre un MBean en un determinado servidor a sus homólogos en los otros servidores del cluster.

Los cambios ha considerar son de dos tipos:

- **Cambio en un atributo:** este es el caso que se produce cuando se cambia una propiedad del MBean (y por lo tanto, del servicio que éste instrumenta).
- **Ejecución de una operación:** cuando se ejecuta una operación de un MBean puede que no se cambie ninguna de sus propiedades pero que sí se realicen acciones que deben ser replicadas en otros servidores del cluster para conservar la homogeneidad del entorno de ejecución.

El servidor de aplicaciones WAS en su versión V. 5.0 usa JMX para monitorizar sus procesos internos y dispone de servicios para realizar notificaciones en su jerarquía y permitir así una administración centralizada.

La jerarquía de servidores en WAS 5.0 es la siguiente:

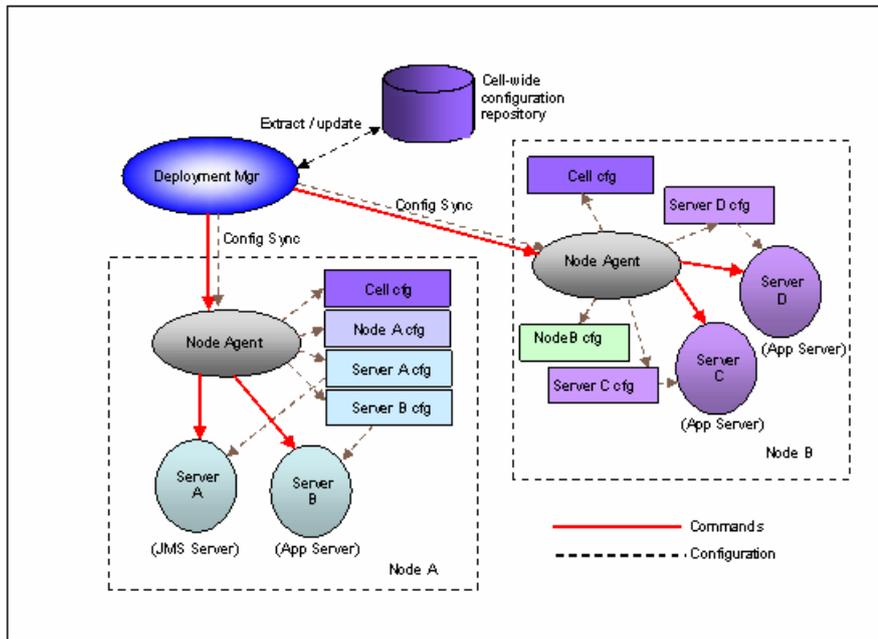


Asumiendo que el actual estándar JMX no tiene contemplado ningún mecanismo para propagar cambios entre MBeans remotos de manera automática, cabría la posibilidad de implementar un mecanismo de notificación entre diferentes MBeans basándonos en los mecanismos que proporciona el servidor Websphere.

Básicamente, la solución consistiría en que, ante la ejecución de un método en un MBean existiese un mecanismo que propagase el cambio a todos los MBeans que cumplieren un determinado perfil.

Para ello, podríamos centralizar en un punto la distribución del evento. El MBean que sufre el evento se lo notificaría al punto de distribución que a su vez sería el encargado de redistribuirlo a los MBeans que cumplan un cierto perfil.

El siguiente gráfico muestra el esquema de funcionamiento que tiene el propio Websphere para notificarse cambios internamente:



Como podéis ver, existe un nodo central que tiene que realizar dos tareas:

- Notificar a los nodos de jerarquía inferior los comandos que él recibe (normalmente desde la consola de administración o desde la herramienta administrativa por línea de comandos)
- Recibir las notificaciones de los nodos de jerarquía inferior y propagarla al resto de nodos. De hecho, los cambios que sufren los nodos (por ejemplo, ha arrancado un servidor) se propagan hacia arriba en la jerarquía.

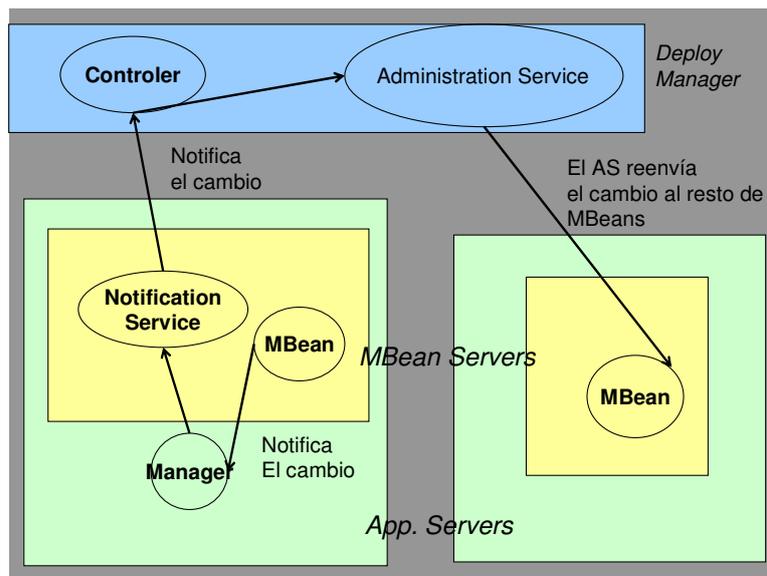
Pues bien, ¿por qué no hacer lo mismo con la gestión de nuestras aplicaciones?. Pensando egoístamente podemos deducir que si la gente de IBM, de BEA, de JBOSS y algunos más han seguido este esquema para su arquitectura de administración puede ser una buena idea hacer lo mismo con nuestros propios componentes.

¿Cómo realizamos cambios y como notificamos los mismos?. El acceso a las propiedades y métodos nos viene dado por la propia naturaleza de los MBeans (ver el proyecto Jakarta-Modeller anteriormente referenciado) y las notificaciones nos la resuelve el servidor de aplicaciones.

En el siguiente apartado se detallan los elementos necesarios para la implementación de una solución de este estilo

2.2 IMPLEMENTACIÓN

El esquema a seguir para la solución propuesta anteriormente se muestra en el siguiente gráfico:



- **MBean:** Los elementos MBean son los elementos que exponen la funcionalidad que se desea instrumentar de los servicios. Los MBean de cada aplicación deberían ser creados por el Manager siguiendo un patrón de factoría basado en el subproyecto Commons-Modeler de Jakarta Commons. Habría que hacer un pequeño desarrollo extra (por que no está realizado en proyecto Modeler de Jakarta) en los MBean para que notificasen al Manager cualquier cambio que sufran.

- **Manager.:** El Manager es la pieza que interacciona con los servicios de tu aplicación y con los MBeans. Cuando un servicio

desea realizar alguna acción con un MBean (crearlo, obtener una referencia,...) debe delegar esta responsabilidad en el Manager.. El Manager no tiene por qué ser un MBean, y puede ser implementado siguiendo el patrón Singleton. Las tareas del Manager son las siguientes:

1. El Manager es el encargado de crear los MBeans y asociarlos a los servicios.
 2. Cuando el Manager crea un MBean, se añade como listener del mismo
 3. Al recibir un evento de cambio o de ejecución de operación realiza las siguientes acciones:
 - o Crea un mensaje con formato XML recopilando los datos necesarios para propagar el evento (propiedad cambiada o método ejecutado, valor de los parámetros,...).
 - o Obtiene una referencia al servicio de notificación de Websphere (el Notification Service) y lo usa para propagar la notificación.
- o **Notification Service:** El Notification Service es un servicio propio de WebSphere. El Notification Service nos va a permitir generar y propagar notificaciones desde elementos que no son MBeans estándares de WebSphere. Este servicio es el que puede usar el Manager para propagar los eventos recibidos de los MBeans.
 - o **Controler:** El Controler es el MBean central que recibe las notificaciones de cambio de los MBeans. Tenemos que desplegar este componente en el Deploy Manager para asegurarnos que va a recibir todas las notificaciones de los nodos de jerarquía inferior. Para recibir y propagar notificaciones, el Controler debe seguir los siguientes pasos:
 1. Registrase como listener al servicio de notificaciones de WebSphere (Notification Service).
 2. Cuando recibe una notificación:
 - o Recupera de la notificación lo que necesite para propagarla (recordemos que la notificación fue creada por nuestro Manager por lo que llevará los datos que nosotros mismos decidamos)
 - o Hace uso del **Administration Service** que es el servicio de WebSphere para propagar los cambios al resto de MBeans del cluster.

2.3 CONCLUSIONES

A modo de resumen sobre instrumentación distribuida se puede decir lo siguiente:

Existe un estándar Java para la instrumentación, el JMX, que está siendo usado por la mayoría de fabricantes de servidores de aplicaciones.

Las necesidades de instrumentación de una aplicación son similares a las que tienen los propios servidores de aplicaciones. ¿Por qué no usar las capacidades que estos servidores nos proporcionan?

Si os interesa este tema podéis contactar conmigo en el mail

spbravo@gmail.com

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

soluciones reales para su negocio

Autentia S.L. Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
 y muchas otras cosas

NUEVO SERVICIO DE NOTIFICACIONES

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
<i>e-mail</i>	
	<input type="button" value="Enviar"/>

OTROS TUTORIALES RECOMENDADOS ([TAMBIÉN VER TODOS](#))

Nombre Corto	Descripción
Struts y EL en Netbeans 4	Os mostramos como configurar la última contribución de Struts que se integra con el lenguaje de expresiones EL.
JDO con OJB	Os mostramos como configurar el entorno OJB de apache para construir la primera aplicación JDO
Arquitectura SOA e Integración de aplicaciones	Os presentamos un breve pero intenso pdf sobre la tecnología Arquitectura SOA y la Integración de aplicaciones
Bibliografía básica recomendada	Os comentamos algunos libros que creemos interesantes para aquellos que quieran avanzar (madurar ideas) en el mundo del desarrollo del Software, a todos los niveles.
Primer vistazo a OpenToro	En este tutorial realizamos una primera inspección (muy positiva) de OpenToro, una herramienta que nos permite realizar la administración y publicación del contenido de tablas de base de datos de un modo rápido y visual.
Analizar ejecución de programa Java	Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.
Cliente y Servidor DCOM con MS Visual C++	Os mostramos como construir un servidor y cliente de Automatización OLE con las facilidades que proporciona el entorno de desarrollo Microsoft Visual C++
Uso de Tiles en Struts	Os mostramos como utilizar el sistema de plantillas proporcionado por Struts (tiles)
Instalación de Together para Eclipse	Os mostramos como instalar la versión de evaluación de Together sobre Eclipse. Estas dos herramientas constituyen una excelente base para el trabajo profesional y serio en entornos Java.
Plantear una aplicación Web y Struts	Os mostramos un posible modo de plantear una aplicación Web (análisis) y darla forma. El FrameWork utilizado es struts y tratamos de identificar qué depende de este FrameWork y qué no.

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.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



**¿Buscas un hospedaje de calidad
por sólo 2 € al mes?**

www.AdictosAlTrabajo.com Opimizado 800X600