



J2SE 6

Novedades

Introducción

- Publicada en Noviembre de 2006.
- Primera versión de Java disponible durante su desarrollo.
- Han colaborado:
 - 160 compañías.
 - 330 desarrolladores de la Comunidad Java.
- Compatible con versiones anteriores.

Web Services

- ¿Qué son?
 - Protocolos y estándares para intercambio de datos entre aplicaciones.
- Java 6 permite exportar una clase como un Servicio Web de manera sencilla.
- Incluye un nuevo parseo para archivos XML.

Web Service. Ejemplo

```
package maths;
```

```
import javax.jws.WebService;
```

```
@WebService
```

```
public class CircleFunctions {
```

```
    public double getArea(double r) {  
        return java.lang.Math.PI * (r * r);  
    }
```

```
    public double getCircumference(double r) {  
        return 2 * java.lang.Math.PI * r;  
    }  
}
```

Web Service. Ejemplo

- Para publicar la clase como un Servicio Web:

```
import javax.xml.ws.Endpoint;
```

```
public static void main(String[] args) {
```

```
    Endpoint.publish(  
        "http://localhost:8080/WebServiceExample/circlefunctions",  
        new CircleFunctions());
```

```
}
```

- El Servicio Web queda publicado en:

– <http://localhost:8080/WebServiceExample/circlefunctions>

Web Services. Ejemplo

- Para utilizar el servicio:

```
public static void main(String[] args) {  
    try {  
        ws.CircleFunctions port = service.getCircleFunctionsPort();  
        double area = port.getArea(2);  
        System.out.println("Area = " + area);  
    } catch (Exception ex) {  
        ex.printStackTrace();  
    }  
}
```

Scripting

- J2SE 6 admite la mezcla de Java con lenguajes de programación dinámicos (PHP, Python, Ruby...) permitiendo:
 - Acceso y control de objetos Java desde un entorno de Script.
 - Crear contenido Web con lenguajes Script.
 - Embeber entornos Script en aplicaciones Java.

- La forma más sencilla de usar este API es:
 - Crear un objeto `ScriptEngineManager`.
 - Obtener un objeto `ScriptEngine` del Manager.
 - Evaluar el Script usando este objeto `ScriptEngine`.

- Ejemplo:

```
import javax.script.*;  
public class Main {  
    public static void main(String[] args) throws Exception {  
        ScriptEngineManager mgr = new ScriptEngineManager();  
        ScriptEngine jsEngine = mgr.getEngineByName("JavaScript");  
        jsEngine.eval("println('hello world')");  
    }  
}
```

Scripting. Ejemplo

```
import java.util.*;
import javax.script.*;

public class Scripting{
    public static void main(String[] args){
        try {
            ScriptEngineManager mgr = new ScriptEngineManager();
            List<ScriptEngineFactory> factories = mgr.getEngineFactories();
            System.out.println("Available script engines:");
            for ( ScriptEngineFactory factory : factories){
                String engine = factory.getEngineName();
                String language = factory.getLanguageName();

                System.out.println("Language: " + language );
                System.out.println("Engine: " + engine);
            }
        }
        catch ( Exception e ) {
            e.printStackTrace();
        }
    }
}
```

Database

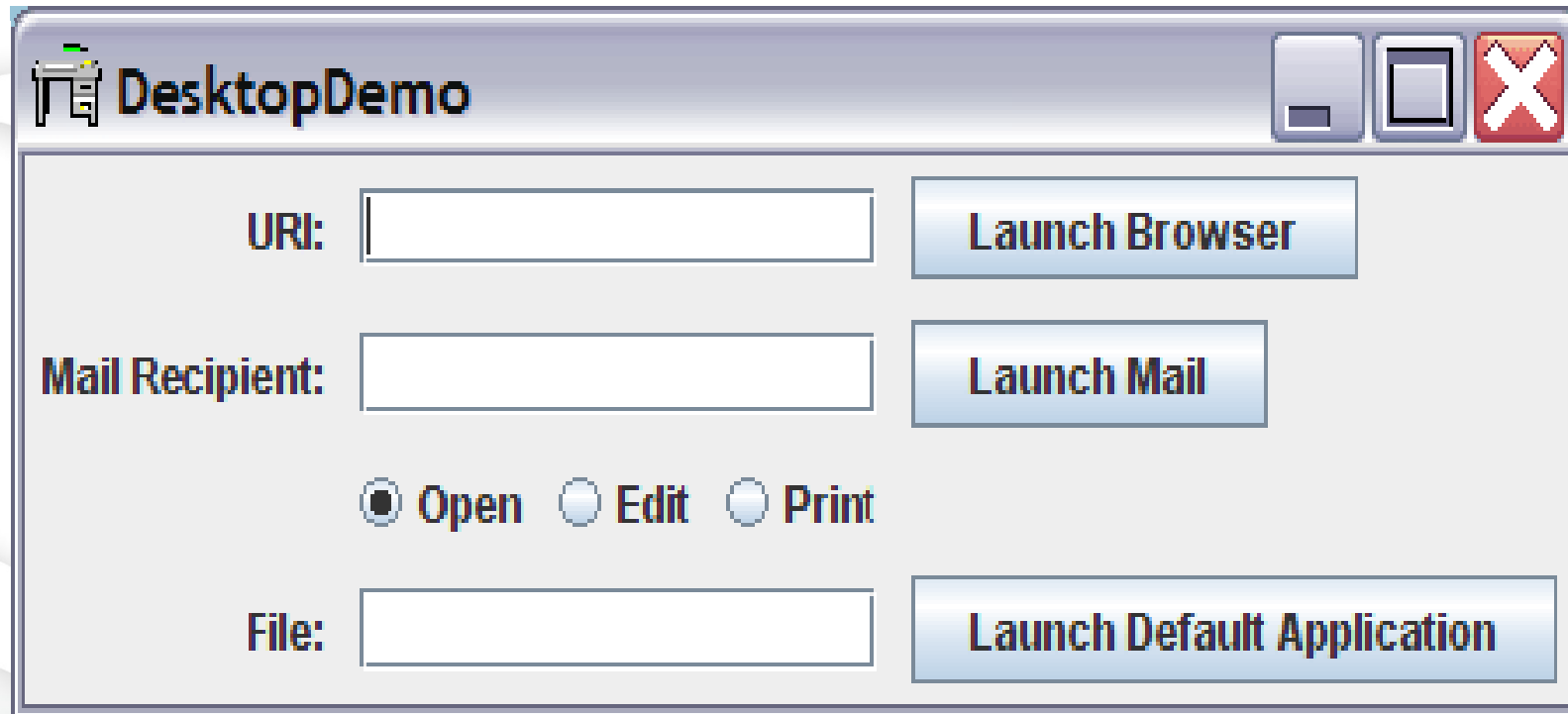
- J2SE 6 incluye en el JDK una Base de Datos.
- Apache Derby de 2MB.
 - Modo cliente-servidor.
 - Modo embebido.
- Funcionalidades:
 - Soporte SQL.
 - Gestión de transacciones.
 - Procedimientos.
 - Triggers.
 - Concurrencia.
 - Backups.

New Desktop APIs

- Nuevo paquete Swing con más funcionalidades.
- Ejemplos:
 - **Splash screen:** pantalla de espera.
 - **System tray support:** adición de iconos, accesos directos y menús desplegables.
 - **Java 2D enhancements:** mejora en la calidad del texto en pantallas LCD.
 - **New java.awt.Desktop API:** eventos de acción Desktop.action integrados en otras aplicaciones.

New Desktop APIs

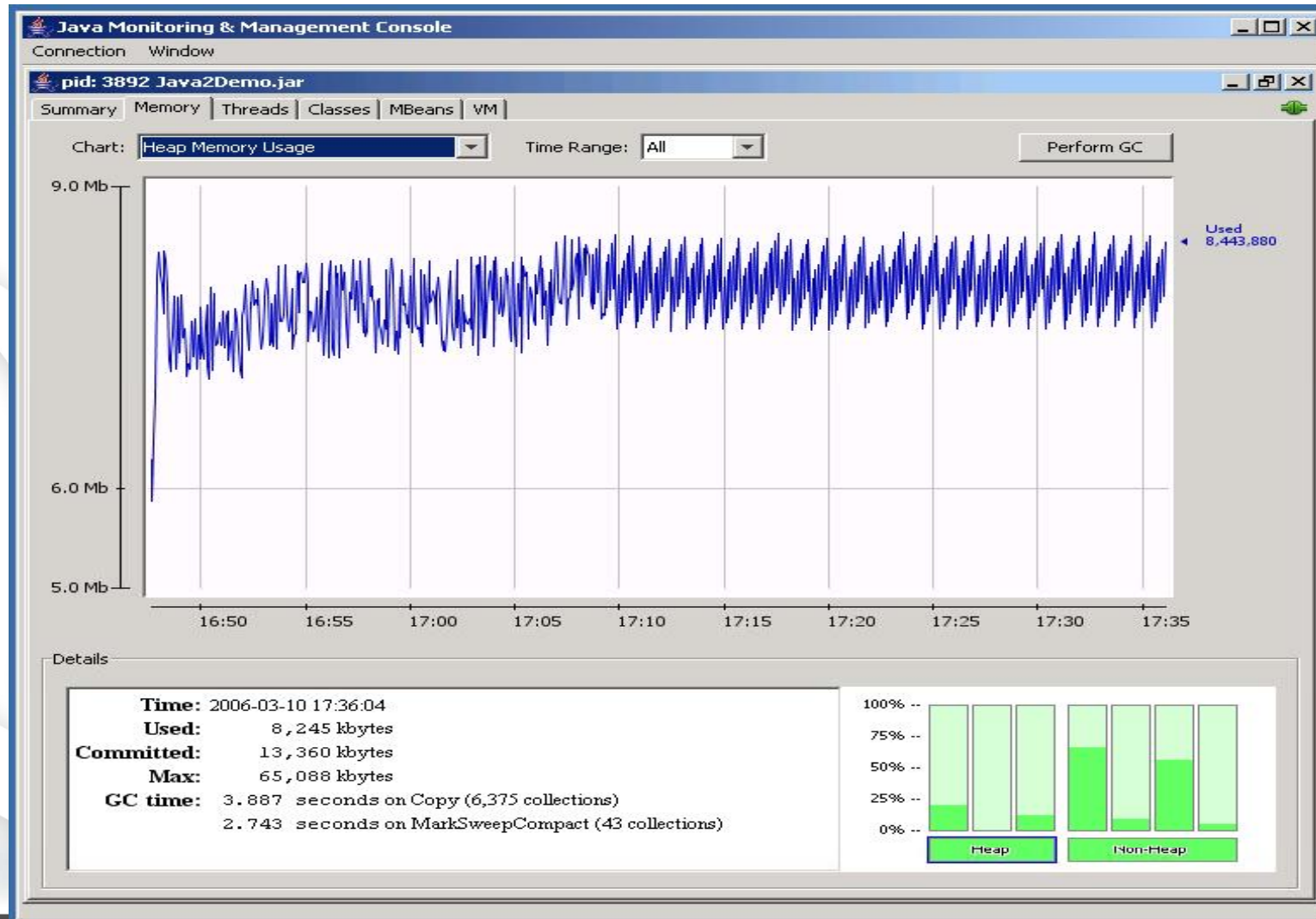
- `Java.awt.Desktop`. Ejemplo:



Monitoring and management

- Herramientas para el control de aplicaciones:
 - **Jconsole**: consola gráfica.
 - **Jps**: procesos.
 - **Jmap & Jhat**: memoria dinámica.
 - **Jstack**: threads.
 - **Jstat**: JVM.

Ejemplo. Jconsole



Compiler Access

- Java Compiler API.
- Tres objetivos:
 - Invocación del compilador.
 - Acceso a mensajes estructurados de diagnóstico.
 - Personalización de ficheros de entrada/salida.
- Ejemplo:

```
//Devuelve un objeto Compiler  
Tool javac = ToolProvider.getSystemJavaCompiler\(\);
```

```
//Lleva a cabo la compilación  
javac.run(null, null, null, arguments);
```

Pluggable Annotations

- Producen un entorno de metaprogramación más poderoso.
- Se pueden usar en clases, métodos y variables.
- Efectos semánticos.
- Por ejemplo, con **@NotNull** especificamos que una variable no puede tomar valores nulos.
- Sin embargo, no cambia el comportamiento en tiempo de ejecución.

Desktop deployment

- Aspecto mejorado de Swing.
- Mayor rapidez en la ejecución de aplicaciones.
- Nueva interfaz de instalación de aplicaciones.
- Nuevas características que mejoran:
 - Experiencia de usuario.
 - Integración en el escritorio.
 - Seguridad.

Security

- API para Firma Digital.
- Mejoras en Java GSS.
- Nuevo servicio de autenticación por Kerberos.
- Accesos a:
 - PKI nativa. Algoritmos de clave pública.
 - Servicios criptográficos de Windows.
 - Servidores LDAP.

Ventajas de J2SE 6

- Compatible con versiones anteriores.
- Ejecutan más rápidamente en el escritorio y en los servidores.
- Más sencilla la localización y resolución de problemas.
- Soporte completo para Windows Vista.

En resumen...

- Plataforma ligera para Web Services.
- JavaScript integrado.
- Soporte XML para BBDD mejorado.
- Monitoreo y diagnóstico de recursos.
- Java DB incluido en el JDK.
- Bibliotecas GUI mejoradas simplifican el desarrollo de aplicaciones cliente enriquecidas.

Y mañana...

- 16:00h
 - Desarrollo de aplicaciones Java EE con Struts 2, Spring y EJB 3.0
- 18:00h
 - Ajax en Java EE.