

Notas técnicas de JAVA Nro. 3 - Tip en detalle

(Lo nuevo, lo escondido, o simplemente lo de siempre pero bien explicado)

Manejo del error “Bad Magic Number”

Tema: Error Bad Magic Number de JAVA

Descripción: Este tip explica las posibles causas del error “Bad Magic Number” y muestra cómo de manera sencilla puede solucionarse este problema.

Nivel: Avanzado

Fecha pub: Octubre 2004

*"Notas Técnicas de JAVA" se envía con frecuencia variable y absolutamente **sin cargo** como un servicio a nuestros clientes. Contiene notas/recursos/artículos técnicos desarrollados en forma totalmente objetiva e independiente. Teknoda es una organización de servicios de tecnología informática y **NO comercializa hardware, software ni otros productos**. Si desea suscribir otra dirección de e-mail para que comience a recibir los tips envíe un mensaje desde esa dirección a develop@teknoda.com, indicando su nombre, empresa a la que pertenece, cargo y país.*

Lista de Tips publicados hasta la fecha:

1. JAVA Basics: Cómo conformar un entorno de programación JAVA (serie de varios tips). Parte I: Selección e instalación de un IDE gratuito.
2. Una introducción a JDBC (Java Database Connectivity) (Acceso a bases de datos desde JAVA)
3. Manejo del error “Bad Magic Number”
4. Java Basics: Entendiendo la Java Virtual Machine

Próximos Tips:

Nivel Técnico avanzado

- Organización de memoria en JAVA Vs. Modelo Tradicional
- JAVA Vs. C++

Nivel Básico

- JAVA Basics: Entendiendo los applets
- JAVA Basics: Entendiendo los servlets
- JAVA Basics: Mitos y Verdades sobre JAVA

Tabla de contenido

El objetivo del tip es explicar la aparición del error Bad Magic Number y como se elimina. Los siguientes puntos serán cubiertos en el mismo :

- I. Introducción
- II. Descripción del error
- III. Recuperación del error
- IV. Información Adicional

I. Introducción

No importa el tiempo que se lleve programando en JAVA, el mensaje de error **Bad Magic Number** es desconcertante, por lo menos cuando se ve por primera vez, y no está bien documentado su origen o sus causas. Se manifiesta en el momento que se intenta ejecutar una clase, por ejemplo, al cargar un applet. No es un error aritmético ni de asignación de tipos, tampoco tiene que ver con algo que uno asocie con las palabras “bad number”, ciertamente el adjetivo “magic” no es muy descriptivo.

Los mensajes en los que por lo general aparece “Bad Magic Number” pueden tener esta apariencia:

- *Exception: java.lang.ClassFormatError nombre_package/nombre_clase (Bad Magic Number)*
- *java.lang.ClassFormatError: (Bad magic number).*

II. Descripción del error

Bad Magic Number se refiere a que una de las clases de nuestro paquete esta corrompida o corrupta. Esta corrupción se debe a ciertos eventos desconocidos que se producen en el momento de compilación, en el momento de la transmisión del código, o por alguna incompatibilidad binaria del bytecode para la JVM encargada de interpretarlo. (Ver al final del artículo los posibles significados del Magic Number).

Una clase corrompida es una clase que por algún motivo no produce el resultado (código saliente) que corresponde y genera por lo tanto la inutilización parcial o total de la misma.

Lo que hay que observar dentro del mensaje de error es lo que a continuación se remarca en color rojo:

- *Exception: java.lang.**ClassFormatError** nombre_package/nombre_clase (**Bad Magic Number**)*
- *java.lang.**ClassFormatError**: (**Bad magic number**).*

Esto nos da la pauta de que pertenece a la Clase **Error** y la misma está relacionada con errores de **compilación**, del sistema o de la **JVM**, por lo tanto son irreuperables.

En la documentación de Sun se puede ver qué ClassFormatError es lanzada cuando la JVM intenta leer un archivo y determina que el mismo está malformado o que no puede ser interpretado como un archivo de clase (.class). La clase anterior “*extiende*” de LinkageError. Subclases de LinkageError indican que una clase tiene cierta dependencia en otras clases; sin embargo, ésta se ha modificado incompatiblemente luego de la compilación de la clase formadora. Además ClassFormatError “*hereda*”, indirectamente, de la clase Error pero aún así no logra ser contemplado mediante: throw o try catch , por ser lanzado en tiempo de ejecución, dado que es el resultado de una situación con condiciones anormales que nunca deberían ocurrir.

III. Recuperación del error

Se aconseja:

- 1- **NO efectuar ninguna modificación en el código y recompilar** varias veces el mismo, hasta que el error desaparezca. (hasta 5 es aceptable)
- 2- Actualizar el JDK.
- 3- Si utiliza applets que estén utilizando .jar, revise que su confección esté bien hecha y que las clases del mismo no estén corrompidas. Antes de formar nuevamente el jar, recompile las clases que el mismo contendrá.
- 4- Si los .class son adquiridos vía FTP, asegúrese que se lo está haciendo en modo binario.
- 5- Si todas las clases están en un mismo archivo .jar, el explorador de internet encuentra el applet pero al intentar cargarlo genera el siguiente error (o similar):

Exception: java.lang.ClassFormatError nombre_package/nombre_clase (Bad Magic Number)

El problema está en que las clases no están direccionadas correctamente. Incluir la clase de Java que maneje los .jar

Detalle de un error de la consola de java:

Error de servidor:

500 Internal Server Error
/techdoc/jsp/TechDoc.jsp:

```
javax.servlet.ServletException: jrun__jsp__TechDoc2ejsp10 (Bad magic number)
java.lang.ClassFormatError: jrun__jsp__TechDoc2ejsp10 (Bad magic number)
at java.lang.ClassLoader.defineClass0(Native Method)
at java.lang.ClassLoader.defineClass(ClassLoader.java:486)
at java.lang.ClassLoader.defineClass(ClassLoader.java:426)
at allaire.jrun.servlet.JSPClassLoader.loadClass(JSPClassLoader.java:82)
at java.lang.ClassLoader.loadClass(ClassLoader.java:253)
at allaire.jrun.servlet.JRunServletLoader.loadServletInstance(JRunServletLoader.java:221)
at allaire.jrun.servlet.JRunServletLoader.loadServletInstance(JRunServletLoader.java:190)
at allaire.jrun.servlet.JRunServletLoader.loadServlet(JRunServletLoader.java:177)
at allaire.jrun.servlet.JRunSE.getServletReference(JRunSE.java:1261)
at allaire.jrun.jsp.JSPServlet.loadServlet(JSPServlet.java:240)
at allaire.jrun.jsp.JSPServlet.loadPage(JSPServlet.java:180)
at allaire.jrun.jsp.JSPServlet.service(JSPServlet.java:168)
at allaire.jrun.servlet.JRunSE.service(JRunSE.java:1013)
at allaire.jrun.servlet.JRunSE.runServlet(JRunSE.java:925)
at allaire.jrun.servlet.JRunRequestDispatcher.forward(JRunRequestDispatcher.java:88)
at allaire.jrun.servlet.JRunSE.service(JRunSE.java:1131)
at allaire.jrun.servlet.JvmContext.dispatch(JvmContext.java:330)
at allaire.jrun.http.WebEndpoint.run(WebEndpoint.java:107)
at allaire.jrun.ThreadPool.run(ThreadPool.java:267)
at allaire.jrun.WorkerThread.run(WorkerThread.java:74)
----->
```

Respecto del significado del término “Magic Number”, se han encontrado definiciones distintas en los foros (ninguna documentada por Sun)

- Magic Number son los primeros cuatro bytes de un archivo de clase (.class) que se encuentran aproximadamente en 0xCAFEBABE en hexadecimal. Esto se puede ver en un editor hexadecimal, levantando cualquier .class. Estos bytes le servirían a la JVM para verificar la integridad de la clase, y que la misma no se vio alterada en alguna etapa del proceso de transmisión.
- Otra definición es cuando un valor representativo no fue asignado a una variable y es utilizado varias veces en distintos puntos del código. Ese valor es llamado Magic Number. Es mejor asignar, en un único punto, un nombre al valor para indicar el significado de ese valor y después utilizarlo a lo largo del programa.

Recordar que siempre lo importante es volver a compilar o armar nuevamente el .jar.

IV. Información Adicional

Sitio de sun: <http://java.sun.com>

Developer Forums <http://forum.java.sun.com>

Copyright 2004 Teknoda S.A. Octubre 2004. JAVA es marca registrada de Sun. SAP, R/3 y ABAP son marcas registradas de SAP AG. AS/400 es marca registrada de IBM. Todas las marcas mencionadas son marcas registradas de las empresas proveedoras.

La información contenida en este artículo ha sido recolectada en la tarea cotidiana por nuestros especialistas a partir de fuentes consideradas confiables. No obstante, por la posibilidad de error humano, mecánico, cambios de versión u otro, Teknoda no garantiza la exactitud o completud de la información aquí volcada.

Dudas o consultas develop@teknoda.com