

# Excelsior Knowledge Base

## PRB: Stack backtrace does not work

Article ID: 000024

Last Revised On: 28-Sep-2005

The information in this article applies to:

- Excelsior JET 4.0 and above

The similar article for Excelsior JET 3.7 and below can be found here [000015](#).

## SYMPTOMS

A JET-compiled application does not print stack backtrace to standard output when an uncaught exception is thrown by one of its threads or when a `printStackTrace()` method is called explicitly. Instead, several "fake" elements are printed:

```
Exception in thread ...
    at java.lang.Void.<unknown>(Unknown Source)
    at java.lang.Void.<unknown>(Unknown Source)
    at java.lang.Void.<unknown>(Unknown Source)
    at java.lang.Void.<unknown>(Unknown Source)
```

## CAUSE

By default, the stack backtrace is disabled in the JET Runtime. Respective methods of the `java.lang.Throwable` class, such as `printStackTrace()`, `getStackTrace()`, `return` and `print` a fake stack trace consisting of `java.lang.Void` elements.

## RESOLUTION

Using the JET Control Panel, you may select the desired level of stack trace support on the **Target** page, in the **Stack Trace Support** combobox. Do not forget to recompile the application after changing this setting.

**Note:** Enabling stack trace may negatively impact performance, if exceptions are thrown and caught repeatedly.

Another way to enable stack trace is to set the `jet.stack.trace` system property, for instance:

```
SET JETVMPROP=-Djet.stack.trace
MyJavaApp.exe
```

It has the same effect as the setting of **Stack Trace Support** to **Minimal** in the JET Control Panel. It enables printing of the stack traces that contain only class names. If you need a detailed information including method names and line numbers, add this option

```
-GENSTACKTRACE+
```

to the beginning of the project file and recompile your application.

**Note:** The side effect of setting GENSTACKTRACE ON is the substantial growth (20% or more) of the resulting executable size.

Enabling the option `-GENSTACKTRACE+` along with the property `jet.stack.trace` is similar to the setting of **Stack Trace Support** to **Full** in the JET Control Panel.

## REFERENCES

1. Excelsior JET User's Guide (<http://www.excelsior-usa.com/doc/jet/jetw.html>), Chapter "Application considerations", section "Java system properties".
2. Excelsior JET User's Guide (<http://www.excelsior-usa.com/doc/jet/jetw.html>), Chapter "Application considerations", section "Stack trace".