

Excelsior Knowledge Base

PRB: Stack backtrace does not work (Excelsior JET 3.7 and below)

Article ID: 000015

Last Revised On: 26-Sep-2005

The information in this article applies to:

- Excelsior JET versions 3.5 through 3.7

This article **does not** apply to Excelsior JET 4.0 and above.

The similar article for Excelsior JET 4.0 and above can be found here [000024](#).

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.

CAUSE

By default, stack trace printing is disabled in the JET runtime system (`Throwable.printStackTrace()` methods do nothing.)

RESOLUTION

To enable stack trace, set the `jet.stack.trace` system property, for instance:

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

Using the Control Panel, you may specify the property on the **Target** page: type `jet.stack.trace` in the **System Properties** field and recompile the application.

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

The `jet.stack.trace` setting enables printing 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.

REFERENCES

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