

## HOWTO: Improve startup time of Java applications

Article ID: 000033

Last Revised On: 05-Jul-2010

The information in this article applies to:

- Excelsior JET 7.2 and above

You may download the latest version of Excelsior JET [here](#).

### INTRODUCTION

You may have noticed that on the first launch after a system reboot (*cold start*), Java applications take longer to start than on subsequent launches (*warm start*), when their files are cached in memory by the operating system. This happens because application startup is I/O bound, so cold startup times heavily depend on the amount and order of disk reads during startup.

Excelsior JET includes the Startup Optimization Toolkit for reducing:

- cold startup time of Java applications by a factor of 1.5x – 3x
- application memory usage by 5–10%

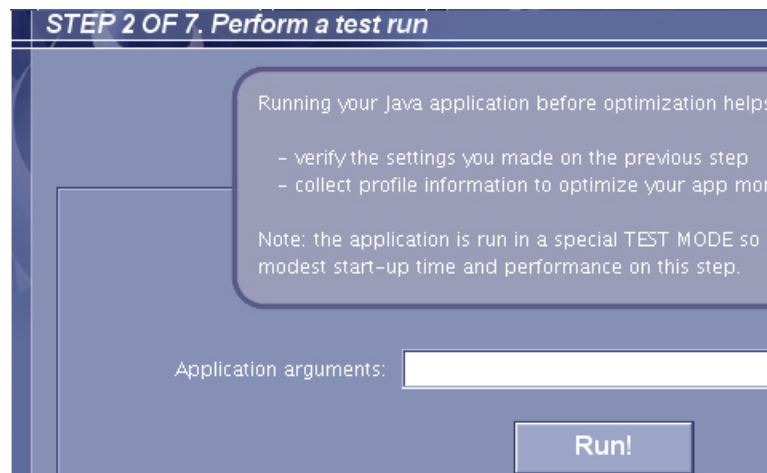
This article explains how to optimize Java application startup.

### DETAILS

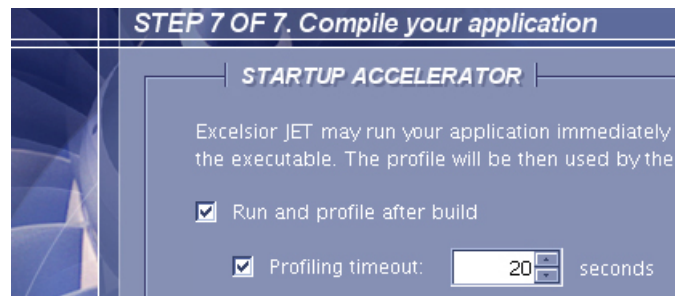
#### How to enable startup optimization

To turn on the optimization, simply run your Java application from within the JET Control Panel to collect profile information.

1. *Before compilation*, run your application at least once on the **Test Run** page of the JET Control Panel.



2. After compilation, the JET Control Panel will automatically launch the compiled application and collect additional profile data. Before starting the build on the Finish page, you may need to adjust the profiling timeout



to ensure the profiling run is long enough to capture application startup.

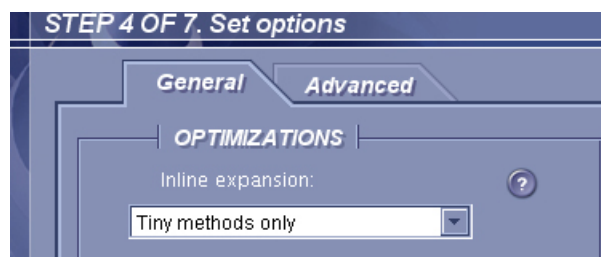
**Note:** Profiling can also be enabled in an unattended build as described in the Excelsior JET User's Guide (the "Startup time optimization" chapter, under "Automated builds")

Once the profile data has been collected, optimization is automatically enabled. You may now reboot the system and evaluate the cold startup time of your Java application.

## How to further reduce the startup time

To achieve the maximum effect of the startup optimization, follow these recommendations:

1. For medium-size applications, such as applications whose classpath includes ten- to fourteen-thousand classes, enable the Global Optimizer to take the standard Java SE packages used by the application in the optimization scope. For more details, see the "Global Optimizer" chapter of the Excelsior JET User's Guide
2. If your software product contains several applications sharing common Java libraries, build the applications together into a multi-app executable. For more details, consult the Excelsior JET User's Guide (the "Application considerations" chapter, under "Multi-app executables").
3. Experiment with the method inlining optimization, which affects the size of the resulting executable, and therefore, may impact startup time: on the Options page of the JET Control Panel,



select the "Low" or "Tiny methods only" inlining policy. Note that this does not necessarily decrease application performance but may improve startup time.

## Known limitations

If you create a multi-component application, that is, compile the application's jar files into the executable and dynamic libraries, the effect of using the Startup Optimization Toolkit may be insignificant because the overhead of loading the libraries is not optimized.

## **FURTHER READING**

The "Startup time optimization" chapter of the JET User's Guide describe this technique in more detail and explains how to integrate the optimization into automated builds.

## **REFERENCES**

1. [Excelsior JET User's Guide](#).