

## Embedded JPF

### [TracNav](#)

- [JPFWiki - Welcome Page](#)

[Introduction...](#)

[Installing JPF...](#)

[User Guide...](#)

[Developer Guide](#)

- [Design](#)
- [Choice Generator](#)
- [Partial Order Reduction](#)
- [Attributes](#)
- [Listener](#)

[MJI...](#)

- [Bytecode Factory](#)
- [Logging](#)
- [Report](#)
- [Embedded](#)
- [JPF tests](#)
- [JPF project layout](#)
- [Create a JPF project](#)
- [Coding Conventions](#)
- [Hosting update site](#)

[Projects...](#)

- [Summer Projects](#)
- [External Projects](#)
- [Change\(B\)log](#)

[About...](#)

- [Events](#)
- [Presentations](#)
- [Papers](#)
- [FAQ](#)
- [History?](#)
- [Support](#)
- [People?](#)
- [Playground](#)
- [Table of Context](#)

JPF can also be used embedded, i.e. called from another Java application. A basic code sequence to start JPF looks like this:

```
import gov.nasa.jpf.JPF;
import gov.nasa.jpf.Config;

void runJPF (String[] args) {
    ..
    MyListener listener = new MyListener(..);
```

```
// [optionally] if you pass through command line args,  
// 'null' any consumed args not to be JPF-processed  
listener.filterArgs( args );  
..  
  
Config config = JPF.createConfig( args );  
// set special config key/value pairs here..  
  
JPF jpf = new JPF( config );  
jpf.addListener( listener );  
jpf.run();  
..  
}
```

Of course, you can also call `gov.nasa.jpf.JPF.main(args)` from within your application, if you don't need to control JPF's configuration or process its output.