

TracNav

- [JPFWiki - Welcome Page](#)

Introduction

- [What is JPF](#)
- [Testing vs model checking](#)
- [Random Example](#)
- [Race Example](#)
- [JPF classification](#)

Installing JPF

- [System requirements](#)
- [Download snapshots](#)
- [Download repositories](#)
- [Create site.properties](#)
- [Install NetBeans IDE plugin](#)
- [Install Eclipse IDE plugin](#)
- [Building and testing](#)

User Guide

- [Application Types](#)
- [JPF Components](#)
- [Configuring JPF](#)
- [Running JPF](#)
- [JPF Output](#)
- [The JPF API](#)

Developer Guide

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

MJI

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

Projects

- [jpf-core](#)
- [jpf-actor](#)
- [jpf-awt](#)
- [jpf-awt-shell](#)

- [jpfc-concurrent](#)
- [jpfc-cv](#)
- [jpfc-delayed](#)
- [jpfc-guided-test](#)
- [jpfc-mango](#)
- [jpfc-racefinder](#)
- [jpfc-rtembed](#)
- [jpfc-statechart](#)
- [net-iocache](#)
- [jpfc-aprop](#)
- [jpfc-numeric](#)
- [jpfc-symbc](#)
- [jpfc-concolic](#)
- [jpfc-symbc-load?](#)
- [jpfc-extended-test-gen](#)
- [jpfc-parallel-spf?](#)
- [eclipse-jpf](#)
- [netbeans-jpf](#)
- [jpfc-inspector](#)
- [jpfc-shell](#)
- [jpfc-template](#)
- [jpfc-trace-server](#)
- [standard NB example](#)
- [Summer Projects](#)
- [External Projects](#)
- [Change\(B\)log](#)

About

- [About this Wiki](#)
- [About the Mailing Lists](#)
- [About the Development Process?](#)
- [About the Repository?](#)
- [How to Contribute](#)
- [JPF contributor account](#)
- [Events](#)
- [Presentations](#)
- [Papers](#)
- [FAQ](#)
- [History?](#)
- [Support](#)
- [People?](#)
- [Playground](#)
- [Table of Context](#)

Running JPF from within NetBeans without JPF plugin

Please Note

The following description assumes NetBeans "freeform projects". In general, it is much better to use the JPF plugins, which not only make the configuration steps described below obsolete, but also work with other NetBeans project types

Since the NetBeans build process is Ant based, and Ant can read a subset of JPF configuration files, integration is fairly straight forward. Every JPF project comes with its own build.xml, and a .../nbproject directory that holds a NetBeans specific project.xml and ide-file-target.xml file, which can serve

as templates for your own projects.

Once everything is configured correctly, you can run JPF by selecting application property (*.jpf) files in the "project view", and do a "Run"->"Run File" from the NetBeans main menu (without the plugin, there will be no "run" item in the project view context menu, but hotkeys should still work).

To make this work, you have to configure two NetBeans specific files:

nbproject/project.xml

If you want to use JPF specific classes (like annotations), you have to make sure the NetBeans source checker will see them. You can achieve this by re-using the JPF property files to set the required classpaths, like

```
...
<general-data xmlns="http://www.netbeans.org/ns/freeform-project/2">
    <name>...</name>
    <properties>
        <property-file>${user.home}/.jpf/site.properties</property-file>
        <property-file>${jpf-core}/jpf.properties</property-file>
        ...
    </properties>
    ...
<java-data xmlns="http://www.netbeans.org/ns/freeform-project-java/2">
    <compilation-unit>
        <package-root>...</package-root>
        <classpath mode="compile">...;${jpf-core.native_classpath}</classpath>
    </compilation-unit>
    ...

```

This works for two reasons:

- Ant supports \${key} expansion from properties (as long as they are not self-recursive)
- Ant properties are single-assignment, i.e. subsequent definitions of the same property name will be ignored

The next step is required no matter if you use JPF types or not - you have to define file actions, which will link to the ide-file-targets.xml

```
...
<ide-actions>
    ...
    <action name="run.single">
        <script>nbproject/ide-file-targets.xml</script>
        <target>run-selected-jpf</target>
        <context>
            <property>jpf.config</property>
            <folder>...</folder>
            <pattern>\.jpf$</pattern>
            <format>absolute-path</format>
            <arity>
                <one-file-only/>
            </arity>
        </context>
    </action>
    ...

```

It seems this has to be defined for each source folder, which can require a number of redundant XML elements

nbproject/ide-file-targets.xml

Within this file, you have to provide the scripts to run the actions defined in the project.xml. This can also make use of the JPF configuration files:

```
<project basedir=".." name="....">
```

```
<property file="${user.home}/.jpf/site.properties"/>
<property file="${jpf-core}/jpf.properties"/>

<path id="base.path">
  ...
  <pathelement path="${jpf-core.native_classpath}" />
</path>
...

<target name="run-selected-jpf">
  <fail unless="jpf.config">Must set property 'jpf.config'</fail>
  <ant antfile="build.xml" inheritall="false" target="compile"/>

  <java classname="gov.nasa.jpf.JPF" failonerror="true" fork="true">
    <arg value="-c"/>
    <arg value="${jpf.config}" />
    <classpath>
      ...
      <path refid="base.path" />
    </classpath>
  </java>
</target>
...
```