

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

jpfc-region

The jpfc-region goal is to provide set of listeners and annotations emulating a new access modifier, which allows programmers to limit access to type, field, method or constructor to specified set of classes. It would come along with Region annotation allowing programmer to gather classes into logical groups.

Description

Confined checker will be used to check whether references to certain type, field or method are used inside of specified scope.

- [Motivating example](#)
- [Confined annotation](#)

Repository

The Mercurial repository is on <http://babelfish.arc.nasa.gov/hg/jpf/jpf-region>

TODOs

There are still some todos for the project:

- correct the project structure: netbeans, lib, tabs, etc
- change AssertionError into PropertyViolation
- refactor all @ConfinedXYZ into one @Confined with different properties
- distinguish between reference and value change checks
- dynamic region has to be explicitly entered and left
- add instance as parameter to @Confined, couple @Confined with @InstanceRegion
- change examples - get rid of TestJPF dependency, run example with verify file
- add support for @Confined in legacy systems using properties system change gov.nasa.jpf.jvm.util to gov.nasa.jpf.aprop.jvm.util

Contact

Filip Rogaczewski, frogaczewski at gmail.com