



链滴

# BTrace 初体验 (一): 新手上路

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1527606917761>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 背景

BTrace是一个java应用动态诊断工具，类似于OpenSolaris平台上的DTrace工具。

BTrace是一个安全无风险的工具，会向运行中的java类注入代码，即"bytecode tracing"。(笔者实测中，发现是有风险的，在BTrace退出时，程序有可能会退出:())

BTrace适用于linux和mac os x平台。

# 安装

## 下载二进制包

直接在[下载页面](#)，下载release包。

## 设置变量

设置**JAVA\_HOME**和**PATH**环境变量

# 使用

## 快速入门

先编写一个示例的java代码，再编写一个示例的BTrace脚本，然后再运行。

## 示例java代码

Looper.java

```
package testBtrace;

public class Looper {

    private static int loopCount = 0;

    private static int loopTime = 0;

    public int loop(int sleepTime) throws Exception {
        System.out.print(
            String.format("sleep %d ms, \tloop count = %d, \ttotal time = %d\n", sleepTime, lo
pCount, loopTime));
        loopCount++;
        loopTime += sleepTime;
        Thread.sleep(sleepTime);
        return loopCount;
    }
}
```

MainLooper.java

```
package testBtrace;

import java.util.Random;

public class MainLooper {
    public static void main(String[] args) throws Exception {
        Looper looper = new Looper();
        Random random = new Random();
        while (true) {
            int sleepTime = random.nextInt(1000) + 1000;
            looper.loop(sleepTime);
        }
    }
}
```

运行效果

```
sleep 1698 ms,   loop count = 0,   total time = 0
sleep 1601 ms,   loop count = 1,   total time = 1698
sleep 1977 ms,   loop count = 2,   total time = 3299
sleep 1597 ms,   loop count = 3,   total time = 5276
sleep 1490 ms,   loop count = 4,   total time = 6873
sleep 1427 ms,   loop count = 5,   total time = 8363
```

## 示例BTrace脚本

在sample目录下有一系列示例脚本，这里选择AllCalls1.java脚本为模板，修改如下：

来跟踪testBtrace.Looper类的调用

```
package com.sun.btrace.samples;

import com.sun.btrace.annotations.*;
import static com.sun.btrace.BTraceUtils.*;

@BTrace public class AllCalls1 {
    @OnMethod(clazz="testBtrace.Looper", method="/*/*",
        location=@Location(value=Kind.CALL, clazz="/*/*", method="/*/*"))
    public static void m(@Self Object self, @TargetMethodOrField String method, @ProbeMethod
        odName String probeMethod) { // all calls to the methods with signature "()"
        println(method + " in " + probeMethod);
    }
}
```

## 运行

设置环境变量

```
export BTRACE_HOME=/path/to/btrace-bin-1.3.11
```

运行

```
[abeffect@note btrace-bin-1.3.11]$ btrace pid AllCalls1.java
valueOf in loop
valueOf in loop
valueOf in loop
format in loop
print in loop
sleep in loop
valueOf in loop
valueOf in loop
valueOf in loop
format in loop
print in loop
sleep in loop
```

退出的方法是：通过Ctrl-c来弹出BTrace控制台，如下：

Please enter your option:

1. exit
2. send an event
3. send a named event
4. flush console output

选择1即可退出.

## 参考

- [btrace github](#)
- [BTrace Wiki](#)
- [神器BTrace快速入门](#)
- [BTrace入门](#)
- [BTrace实战](#)