

Guide to the MMTk refactoring

Robin Garner
Daniel Frampton

August 10, 2005

1 Introduction

This is a guide to the recent MMTk refactoring, intended to help people who have existing MMTk plans to refactor these plans to fit the new structure.

2 The new structure

The most obvious change is that an MMTk plan has now been split into 3 or more classes. The main aim of this refactoring is to enable the use of instance methods for methods that used to be static, and hence make better use of inheritance to structure a plan.

Previously, system-global data was kept in static fields of a plan, and thread-

2.2 Tracing

The core of most garbage collections is the operation of tracing the heap. This

5. Create a collectionPhase method - this takes 2 more parameters than the global equivalent.

```
public final void collectionPhase(int phaseId, boolean participating,  
                                boolean primary)
```

```
throws InlinePragma {
```

```
if (phaseId == MS.PREPARE) {
```

```
    super.collectionPhase(phaseId, participating, primary);
```

```
    // Contents of threadLocalPrepare()
```

```
    trace.prepare();
```

```
return(;)J109d[()]J009d[(R)2E)0)0(J109d[s)0u)0p)0e)0r)0.)5c)0o)0de)00(u)0u)0e
```

```
return(;)J109d[()]J009d[(R)2E)0)0(J109d[s)0u)0p)0e)0r)0.)5c)0o)0de)00(u)0u)0e
```

