

RAGE

White-Papers

Kernel

Software & Documentation © The RAGE Team 2005

Version Relavence: JGE Release 0.2 + JGF Future Release

CONTENTS

Overview	3
Where is the Application?	4
High-Level Diagram	5

OVERVIEW

What is the Kernel?

The Kernel is defined by anything that is essential for any RAGE application to run. It defines all the basic core systems that are depended on by nearly every other RAGE subsystem. Therefore, every application written using RAGE will include the Kernel.

There are four main systems provided by the Kernel. These are:

- Task Management
- Low Level Timing
- Resource Management
- Properties Management

Task Management

Virtually every RAGE application will be implemented using RAGEs Task based execution paradigm. An application will extend the `Task` superclass, and then have the Kernel's `TaskManager` execute the `Task` (along with several other system tasks). The `TaskManager` also provides some basic profiling facilities, and can provide information on how long each task is taking to execute compared to the total update time. The Task Management system is packaged in `jge.kernel.task`.

Low Level Timing

The Kernel provides a base `Timer` class, providing a common interface to low level timing. Timers in RAGE are accurate to the nanosecond (i.e, to the nearest billionth of a second). Because of the inaccurate nature of the Java timer (as of 1.4.2), timers are usually implemented at the native level (i.e, with operating system dependant code). Hopefully, future versions of Java will provide a more accurate timing method so as to eliminate the need for this system level dependancy. The timing system is packaged in `jge.kernel.timer`.

Resource Management

The Resource Management system provided by the Kernel acts as an easy mechanism to load external data and convert the data into a particular object. The `ResourceManager` has the capability to load files from a local disk or from a remotely from a http server. The `ResourceManager` also acts as a local cache for data, so as to eliminate unnecessary loading of the same data twice.

The `ResourceManager` has no direct support for any file formats, to completely eliminate any dependancy. Support for file formats is be provided by derived `ResourceLoaders`. The Resource Management system is packaged in `jge.kernel.resource`.

Properties Management

The Kernel provides a central repository for system and application properties. Properties are essentially managed global variables. The properties management system allows classes to create and modify properties, as well as to observe changes in the properties manager. The properties manager also provides a safe mechanism for loading properties from a local file. The Properties management system is packaged in `jge.kernel.properties`.