

Home

The screenshot shows the MoSKito WebUI interface. It features a navigation bar with tabs for Producers, Journeys, Thresholds, Accumulators, Threads, and More. Below the navigation bar, there are several data tables. The first table is titled 'MemoryPool' and lists various memory pools with columns for Producer Id, Category, Subsystem, Free, Free MB, Init, Init MB, Min Used, Min Used MB, Used, and Used MB. The second table is titled 'ThreadCount' and lists thread counts with columns for Producer Id, Category, Subsystem, Started, Min Cur, Current, Max Cur, Daemon, and class. The third table is titled 'ThreadStates' and lists thread states with columns for Producer Id, Category, Subsystem, Min, Current, Max, and class. The fourth table is titled 'VirtualMemoryPool' and lists virtual memory pools with columns for Producer Id, Category, Subsystem, Free, Free MB, Init, Init MB, Min Used, Min Used MB, Used, Used MB, and Max Used.

Welcome to MoSKito Confluence Space!

MoSKito is an open source system for monitoring Java web applications.

With MoSKito, you may:

- collect, store and analyse any type of performance data (MoSKito-Essential),
- create personalised storage for the collected data (MoSKito-Central),
- effectively monitor multi-node applications (MoSKito-Control).

In fact, MoSKito is a complete system kit for DevOps (as well as classical Devs and Ops) who care about performance of their web apps.

On this Confluence Space, you will find all the existing MoSKito

documents. Browse the sections below, navigate the Confluence sidebar or use Search to find the documents you currently need.

Enjoy!

Anotheria Team

MoSKito Projects and Components

Within MoSKito system, we differentiate between **projects** and **components**:

- **Projects** can be released as standalone applications,
- **Components** are parts (modules in *maven* sense) of a project.

Every section below represents an existing MoSKito Project.

MoSKito-Essential

is the heart of MoSKito framework, a system kit for web application monitoring.

MoSKito-Essential is enough to watch a single-node (having one JVM) application. With MoSKito-Essential, you may collect the widest range of performance data, store some of that data and instantly analyse it.

MoSKito-Essential section

MoSKito-Central

is a centralised data storage server for keeping performance snapshots, taken by MoSKito.

MoSKito-Essential monitors here and now, without storing the collected data. With MoSKito-Central, you may setup a local data warehouse, the way you need, and store performance snapshots perfectly organised.

[MoSKito-Central section](#)

MoSKito-Control

is a tool for monitoring multi-node web applications, managing info from multiple MoSKito instances.

MoSKito-Control is a 'control centre' for your distributed app: it gathers info from multiple MoSKito instances and displays the health state of the whole application, as a sum of its components.

[MoSKito-Control section](#)