

Home

The screenshot shows the MoSKito WebUI interface with several data tables. The 'MemoryPool' table lists various memory pools with their categories, subsystems, and usage statistics. The 'ThreadCount' table shows thread counts for different subsystems. The 'ThreadStates' table shows the distribution of thread states. The 'VirtualMemoryPool' table shows virtual memory pool usage statistics.

Producer Id	Category	Subsystem	Free	Free MB	Init	Init MB	Min Used	Min Used MB	Used	Used MB
MemoryPool-Code Cache-NonHeap	memory	builtin	180,096	0	50,331,648	48	2,516,544	2	15,483,008	14
MemoryPool-PS Eden Space-Heap	memory	builtin	74,252,264	70	168,886,272	161	24	0	68,026,392	64
MemoryPool-PS Old Gen-Heap	memory	builtin	49,592,648	47	447,414,272	426	74,220,736	70	185,747,128	177
MemoryPool-PS Perm Gen-NonHeap	memory	builtin	702,480	0	134,217,728	128	30,258,264	28	76,761,072	73
MemoryPool-PS Survivor Space-Heap	memory	builtin	34,880	0	18,874,368	18	0	0	18,839,488	17

Producer Id	Category	Subsystem	Started	Min Cur	Current	Max Cur	Daemon	class
ThreadCount	threads	builtin	5,863	27	70	75	68	BuiltinThreadCountProducer

Producer Id	Category	Subsystem	Min	Current	Max	class
ThreadStates	threads	builtin	27	70	75	BuiltinThreadStatesProducer

Producer Id	Category	Subsystem	Free	Free MB	Init	Init MB	Min Used	Min Used MB	Used	Used MB	Max Used
Heap memory	memory	builtin	123,879,792	118	635,174,912	605	74,220,760	70	272,613,008	259	440,776,792
Non-heap memory	memory	builtin	882,576	0	184,549,376	176	32,774,808	31	92,244,080	87	92,244,080

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](#)