[EN] 02. The Functional Scope of the Administrative Interface of the Server

In order to connet to the remote interface made available in the JMX technology, it is usually necessary to enter the address of the interface, specially formatted, and the authentication details. In the case of the dLibra system, the address has the following form: **service:jmx:rmi:///jndi/rmi://[adres serwera]:[port serwera]/mx** The authentication data are the username and password for the 'admin' account defined in the *dLibra* server.

In the image below, the administrative interface of the *dLibra* system, managed with the help of the JConsole program, is shown. The tab with the general view makes it possible to monitor the usage of the server and memory, the number of threads, and the loaded classes. The details are presented in separate (following) tabs.
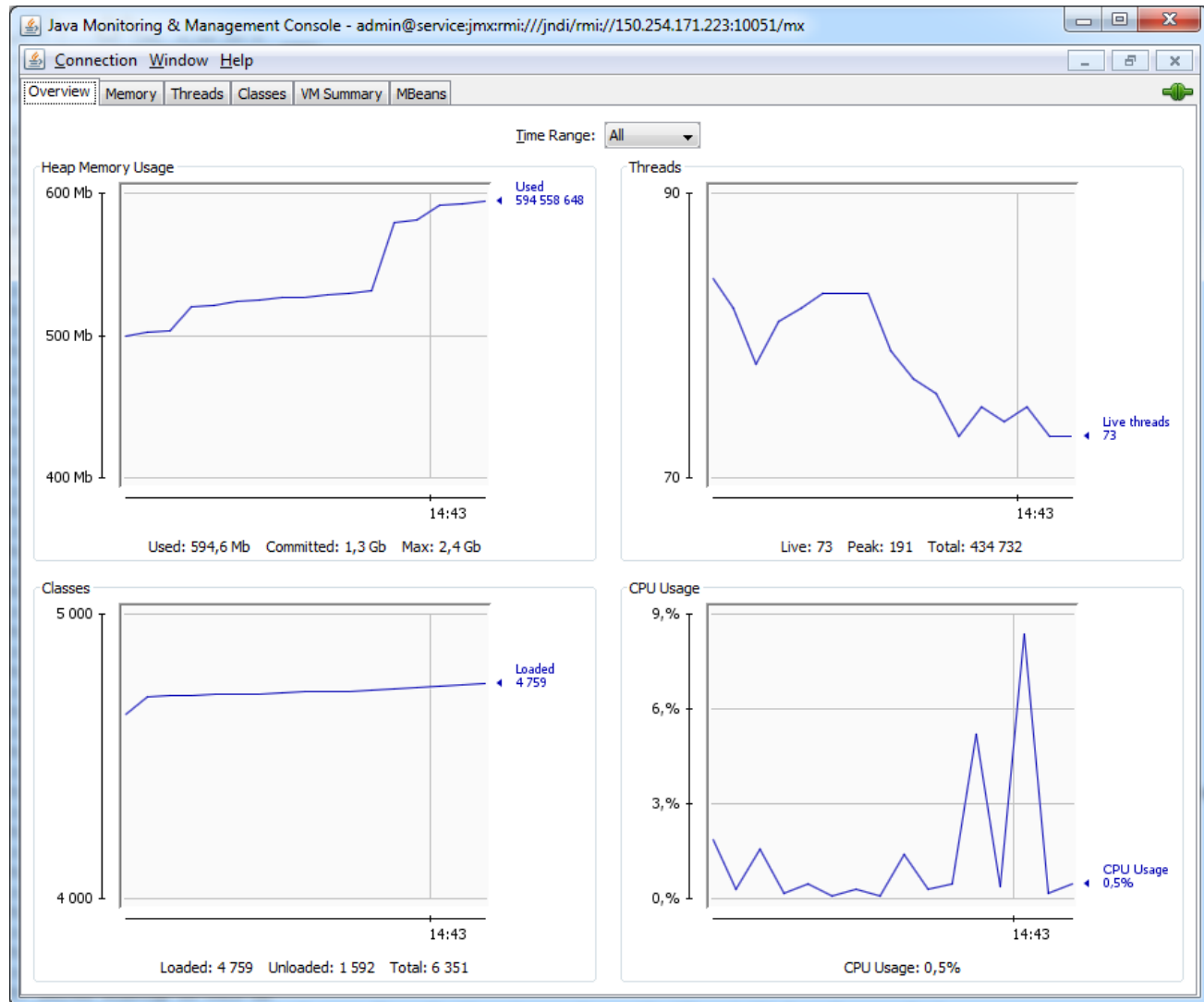
Apart from the standard management components made available by the environment of the Java virtual machine, the dLibra system has dedicated components for server manament. They are in the last tab, which is called MBeans. Those components are divided into groups which correspond to the internal structure of the server. Those groups are further divided by function. The result is a hierarchical, tree-type structure of components. According to the JMX specification, each component can make properties and methods available. Below, the most important elements are described. The omitted elements are, for the most part, only useful in exceptional circumstances. They require in-depth knowledge of the internal structure of the *dLibra* system and should not be used. The following notation is used in the description:

- *NazwaAtrybutu* – the name of an attribute, the value of which can be read and (optionally) modified; and
- *nazwaMetody(parametr1,parametr2)* – a method (command) which can be called. Non-parameter methods are also possible.
  In general, attributes can be used for monitoring the operations of the server, and methods can be used for, for example, carrying out certain maintenance works, migrating data between version, or recovering some data lost because of a failure.

> **⚠ Warning**
>
> In the case of some components, it is relatively easy to make significant changes in the data stored in the digital library. Some of those operations may be very time-consuming – for example, a full re-indexation of the publications of a large digital library may take, depending on the efficiency of the server, up to a few weeks. **The greatest caution is recommended when the administrative interface of the dLibra server is used.**

Content Server

ZipCache:

- *CacheTimeSpan* – the life time in the cache of the oldest edition (in hours),
- *CurrentZipCacheSize* – the actual cache size (in bytes),
- *CurrentZipCacheSize* – the actual cache size in relation to the maximum size (as a percentage),
- *MaxZipCacheSize* – the maximum cache size (in bytes),
- *NoOfCachedZipFiles* – the number of files in the ZIP format in the cache,
- *NoOfZipCacheHits* – the number of 'hits' in the cache,
- *NoOfZipCacheMisses* – the number of 'misses' in the cache,
- *OldestAccessDate* – the date of the access to the oldest file in the cache, and
- *ZipCacheHitRatio* – the ratio of the 'hits' in the cache to the general number of requests.

Event Manager

EventMonitoringBean:

- *NoOfEveEvents* – shows the number of events which are yet to be handled by the services of the dLibra server; an event can be adding a new publication or modifying a description, and the handling of it may consist in, for example, updating search indexes; the number of tasks should go down to zero at least once a day (or once a week); if that is not the case, it means that the *dLibra* server is overworked and is not capable of handling the tasks generated by library users on an ongoing basis;
- *EventProcessingDelay* – an event processing delay (in hours); and
- *OldestEventDate* – the date of the oldest unhandled event.

Metadata Server

MaintenanceTasks:

- *reloadDCPatterns()* – reloads the information about mapping the attribute set defined in the digital library to the DublinCore schema; the information is in the *dc-pattern.properties* file, in the conf subdirectory of the root directory of the server;
- *expireAndDeleteTags()* – runs the task of user tag checking, which makes tags older than 30 days obsolete and removes tags older than 60 days; that task is also carried out cyclically by the server; and
- *normalizeAttributeValues(String rdfName, Integer type)* – normalizes values for the given attributes and for the particular type. The argument takes the following values:
  - **type**
    - 1 – normalization to the date form
    - 2 – normalization to the language form
    - 3 – normalization to a form of a particular type

Search Server

IndexBackupTasks:

- *backupIndex(mode)* – causes the creation of a backup copy of an index, depending on the given parameter values, which can be as follows:
  - **mode:**
    - 0 – does not create a backup copy of any index,
    - 1 – creates a backup copy of the main index,
    - 2 – creates a backup copy of the synonym index,
    - 3 - 1 i 2,
- *restoreIndex(mode)* – removes the current indexes and recreates them from a backup copy, depending on the given parameter values; the meaning of parameters is the same as in the *backupindex* function; and
- *synchronizeIndices(mode)* – synchronizes the current search indexes with the indexes used by the Index Server, depending on the given parameter values; the meaning of parameters is the same as in the *backupindex* function.

Index Server

IndexBackupTasks:

- *backupIndex(mode)* – causes the creation of a backup copy of an index, depending on the given parameter, which have the following values:
  - **mode:**
    - 0 – does not create a backup copy of any index,
    - 1 – creates a backup copy of the main index,
    - 2 – creates a backup copy of the synonym index,
    - 3 – 1 and 2; and
- *restoreIndex(mode)* – removes the current indexes and recreates them from a backup copy, depending on the given parameter values; the meaning of parameters is the same as in the *backupindex* function.

ReindexTasks:

- *reindexAll(indexMode)* – causes a reindexation of the whole content of the digital library;
  - **indexMode:**
    - 0 – does not carry out the reindexation,
    - 1 – reindexes editions (metadata and resources),
    - 2 – reindexes publications (planned publications, group publications, metadata),
    - 4 – indexes synonyms,
    - 7 – 1, 2, and 4;
- *checkIndexesConsistency(indexMode, fix)* – checks and corrects the consistency of search indexes; parameters:
  - **indexMode** – functions in the same way as in the case of reindexAll;
  - **fix**:
    - true – corrects inconsistencies,
    - false – leaves inconsistencies uncorrected; and
- *reindexEditions(editionsIds)* – causes a reindexation of editions with the given identifiers; the value of parameter **editionsIds** should indicate the identifiers of the editions which are to be indexed, comma-separated.

dLibra JMX Management Service

RMIConnectionsInfo:

- *NoOfRMIThreads* – the number of external connections to the *dLibra* server; and
- *RMIConnectionIPs* – the IP addresses of the computers from which connections to the server are being made.

ServiceWrapperManager:

- *stop()* – switches off the server; and
- *restart()* – restarts the server.