

Podręcznik administratora serwera dLibra

Poznańskie Centrum Superkomputerowo - Sieciowe
Projekt dLibra



Redakcja: Marcin Werla

Podręcznik administratora serwera *dLibra*

Marcin Werla

Copyright © 1999 - 2007 Poznańskie Centrum Superkomputerowo - Sieciowe

Spis treści

Wstęp	iv
1. Dla kogo przeznaczona jest ta instrukcja?	iv
2. Co zawiera ta instrukcja?	iv
3. Dodatkowe źródła informacji	iv
1. Konfiguracja	1
2. Panel administracyjny	4
2.1. Budowa interfejsu administracyjnego serwera systemu dLibra	4
2.1.1. Technologia JMX	4
2.1.2. JConsole	4
2.1.3. MC4J	4
2.2. Zakres funkcjonalny interfejsu administracyjnego serwera	5
2.2.1. Content Server	7
2.2.2. Event Manager	7
2.2.3. Metadata Server	7
2.2.4. OAI-PMH Harvester	8
2.2.5. Search Server	9
2.2.6. dLibra JMX Management Service	9

Wstęp

1. Dla kogo przeznaczona jest ta instrukcja?

Instrukcja ta jest przeznaczona dla osób, które chcą poznać możliwości konfiguracyjne oraz mechanizmy administracyjne serwera *dLibra*. Ma ona pomóc w administracji i utrzymaniu systemu.

Aby być w stanie zarządzać biblioteką cyfrową opartą o oprogramowanie *dLibra*, użytkownik powinien posiadać wiedzę i umiejętności wystarczające na sprawne poruszanie się w systemie plików systemu operacyjnego, w którym biblioteka cyfrowa będzie uruchomiona. Powinien on również być w stanie obsługiwać dodatkowe oprogramowanie wymagane do działania systemu *dLibra*. Niezbędna jest również znajomość języka angielskiego w stopniu pozwalającym na korzystanie z anglojęzycznej dokumentacji technicznej.

2. Co zawiera ta instrukcja?

Instrukcja ta składa się ona z następujących rozdziałów:

Rozdział 1. *Konfiguracja* Zawiera omówienie poszczególnych plików konfiguracyjnych serwera.

Rozdział 2. *Panel administracyjny* Opisuje panel administracyjny serwera realizowany w oparciu o technologię Java JMX.

3. Dodatkowe źródła informacji

Osoby poszukujące dodatkowych informacji na temat systemu *dLibra* powinny skierować się m.in. na następujące strony:

- <http://dlibra.psnc.pl/> - Oficjalna strona projektu *dLibra*. Zawiera m.in. bazę wiedzy na temat projektu.
- <http://dlibra.psnc.pl/forum/> - Forum użytkowników projektu *dLibra*.
- <http://dlibra.psnc.pl/biblioteka/> - Biblioteka cyfrowa projektu *dLibra*. Zawiera oficjalną dokumentację systemu oraz wiele artykułów i prezentacji na temat systemu.

Osoby zainteresowane tematyką digitalizacji i bibliotek cyfrowych zachęcamy też do odwiedzenia tematycznego forum dyskusyjnego czasopisma EBIB, poświęconego digitalizacji. Forum to dostępne jest pod adresem: <http://ebib.oss.wroc.pl/phpBB/viewforum.php?f=7>. Polecamy również strony Biblioteka 2.0 stanowiące forum społeczności bibliotekarzy i czytelników cyfrowych.

Rozdział 1. Konfiguracja

Uwaga!

Zaleca się zachowanie szczególnej ostrożności przy modyfikowaniu wszystkich opisanych poniżej plików i parametrów systemu *dLibra*.

Wszystkie pliki konfiguracyjne systemu *dLibra* znajdują się w podkatalogu `conf` katalogu głównego serwera oraz jego podkatalogach. W każdym z plików istotne parametry konfiguracyjne zostały opatrzone komentarzem. Pliki znajdujące się bezpośrednio w katalogu `conf` dotyczą wszystkich usług serwera systemu *dLibra*. Pliki znajdujące się w podkatalogach tego katalogu zawierają indywidualne ustawienia poszczególnych usług. Dostępne są następujące pliki konfiguracyjne:

- `database.properties` - zawiera nazwę użytkownika, hasło oraz adres bazy danych wykorzystywanej przez serwer *dLibra* oraz parametry definiujące rodzaj wykorzystywanej bazy danych.
- `default-service.properties` - zawiera domyślne wspólne parametry konfiguracyjne usług serwera *dLibra*. Zdefiniowane w tym pliku wartości mogą być nadpisywane w plikach `service.properties` poszczególnych usług.
- `diacritic.properties` - plik określający zasady transformacji znaków diakrytycznych na potrzeby indeksowania i wyszukiwania.
- `ehcache.properties`, `*.dtd` - pliki konfiguracyjne mechanizmu dostępowego do bazy danych.
- `log4j.properties` - plik konfigurujący mechanizm logowania przebiegu pracy serwera.
- `lucene.properties` - plik konfigurujący mechanizm indeksowania i wyszukiwania.
- `server.xml` - główny plik konfiguracyjny serwera. Zawiera on listę usług serwera *dLibra*, które mają być uruchomione oraz wskazania na pozostałe pliki konfiguracyjne.
- `smtp.properties` - plik konfigurujący mechanizm wysyłania wiadomości e-mail (wykorzystywany opcjonalnie, w zależności od konfiguracji mechanizmu logowania).
- `stopwords.properties` - plik zawierający listę plików ignorowanych podczas indeksowania i wyszukiwania.
- `wrapper/*.conf` - zestaw plików konfiguracyjnych mechanizmu odpowiedzialnego za uruchomienie serwera *dLibra* w trybie usługi. Szczegółowa dokumentacja tego pliku znajduje się na stronie <http://wrapper.tanukisoftware.org/doc/english/-properties.html>. Można tu m.in. ustawić maksymalne zużycie pamięci przez serwer *dLibra* (parametr `wrapper.java.maxmemory` w pliku `base.conf`).
- `??/quartz.properties` - plik konfigurujący mechanizm zadań okresowych.
- pliki `??/jobs.xml` - pliki konfigurujące zadania okresowe na poziomie poszczególnych usług serwera *dLibra*.
- `??/service.properties` - plik zawierający parametry konfiguracyjne specyficzne dla poszczególnych usług.

- `cs/zip_readme.txt` - szablon informacji tekstowej dołączanej do archiwów z treścią publikacji.
- `ms/dc-export.properties` - plik określający zasady transformacji metadanych do schematu DublinCore.
- `mx/xls` - katalog zawierający pliki wykrozystywane do generowania interfejsu administracyjnego serwera.
- `se/liusConfig.xml` - plik konfigurujący mechanizm indeksujący treść publikacji.
- `us/ldap.properties` (plik opcjonalny) - zawiera zaawansowane parametry konfiguracyjne autoryzacji opartej o LDAP, wykorzystywane do obsługi mechanizmu referrals LDAP. Ma on następującą strukturę:

```
ldap.server.host.l=host
ldap.server.base.dn.l=baseDN
ldap.admin.dn.l=adminDN
ldap.admin.password.l=password
...
ldap.server.host.n=host
ldap.server.base.dn.n=baseDN
ldap.admin.dn.n=adminDN
ldap.admin.password.n=password

domain.search.prefix.l=prefix
domain.name.l.l=domain
...
domain.name.l.m=domain
...
domain.search.prefix.n=prefix
domain.name.n.l=domain
...
domain.name.n.m=domain
```

Znaczenie wpisów jest następujące:

- **ldap.server.host.n** - adres referralu
- **ldap.server.base.dn.n** - baseDN od którego powinno zaczynać się szukanie wpisów
- **ldap.admin.dn.n** - pełny DN użytkownika administracyjnego na danym referralu,
- **ldap.admin.password.n** - hasło użytkownika administracyjnego na danym referralu

Podstawowym wpisem w tej grupie jest adres referralu i jeśli go nie będzie to pozostałe wpisy z tą samą liczbą na końcu będą zignorowane. Jeśli natomiast nie będzie któregoś z pozostałych wpisów, to zostanie przyjęta domyślna wartość taka jak dla serwera głównego. Nie ma ograniczenia na liczbę referrals n.

Wpisy związane z obsługą domen mają następujące znaczenie:

- **domain.search.prefix.n** - prefiks, który będzie dołożony przed baseDN serwera głównego

- `domain.name.n.m` - domena dla której prefiks ma być doklejony

Może być dowolna ilość grup wpisów, gdzie dla danego prefiksu definiujemy dowolną ilość powiązanych z nim domen. Należy zwrócić uwagę, że prefiks jest doklejany do baseDN serwera głównego - wynika z tego ograniczenie, że sufiks na referralach musi być taki jak podane dla głównego serwera baseDN.

Rozdział 2. Panel administracyjny

2.1. Budowa interfejsu administracyjnego serwera systemu dLibra

2.1.1. Technologia JMX

Do budowy interfejsu administracyjnego serwera systemu *dLibra* wykorzystana została technologia Java JMX . Technologia ta umożliwia dostęp do zdalnych obiektów języka Java, przeglądanie i modyfikację własności tych obiektów oraz wywoływanie operacji (metod) udostępnianych przez te obiekty. Udostępniane obiekty mogą być grupowane w hierarchiczne struktury. Funkcjonalność technologii Java JMX może być udostępniana poprzez wiele protokołów takich jak HTTP, WebServices czy SNMP. W projekcie *dLibra* zdecydowano się na wykorzystanie protokołu Java RMI. Szczegółowe informacje na temat Java JMX znaleźć można na oficjalnej stronie z dokumentacją oraz w wielu artykułach pokazujących wykorzystanie tej technologii.

Aby móc skorzystać z obiektów udostępnianych przy pomocy interfejsu Java JMX należy posiadać oprogramowanie klienckie wspierające tą technologię. Poniżej opisano dwa takie programy: JConsole i MC4J. Obydwa są bezpłatne i pozwalają na wykorzystanie możliwości jakie daje interfejs administracyjny serwera systemu *dLibra* . Inne programy tego typu to np.: EJTools JMX Browser , Panoptes , JManage .

Dodatkowo poprzez odpowiednie ustawienie wpisów w pliku konfiguracyjnym `conf/mx/service.properties` możliwe jest uruchomienie interfejsu administracyjnego serwera opartego o strony WWW wbudowanego w serwer *dLibra*. Interfejs ten daje dostęp do podstawowych możliwości panelu administracyjnego serwera.

2.1.2. JConsole

JConsole jest programem dostarczonym wraz z maszyną wirtualną Javy w wersji 1.5. Znajduje się on w podkatalogu `bin` katalogu, w którym zainstalowano maszynę wirtualną.

Aby podłączyć się do serwera *dLibra* należy w okienku "Connect to Agent" (menu Connection|New connection...) wybrać zakładkę Advanced i wprowadzić tam dane zgodnie z opisem konfiguracji połączenia przedstawionym poniżej.

Oficjalna dokumentacja JConsole znajduje się pod adresem <http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html> . Firma Sun Microsystems Inc. nie zaleca korzystania z programu JConsole w środowiskach produkcyjnych. Program ten jest raczej przeznaczony do zastosowań testowych i programistycznych.

2.1.3. MC4J

MC4J jest darmowym oprogramowaniem umożliwiającym efektywne wykorzystanie interfejsów JMX. Ze względu na swoje rozbudowane możliwości jest on obecnie oprogramowaniem zalecanym do korzystania z interfejsu administracyjnego serwera *dLibra* .

Aby podłączyć się do serwera *dLibra* należy w pierwszym kroku kreatora nowego połączenia wybrać następujące wartości:

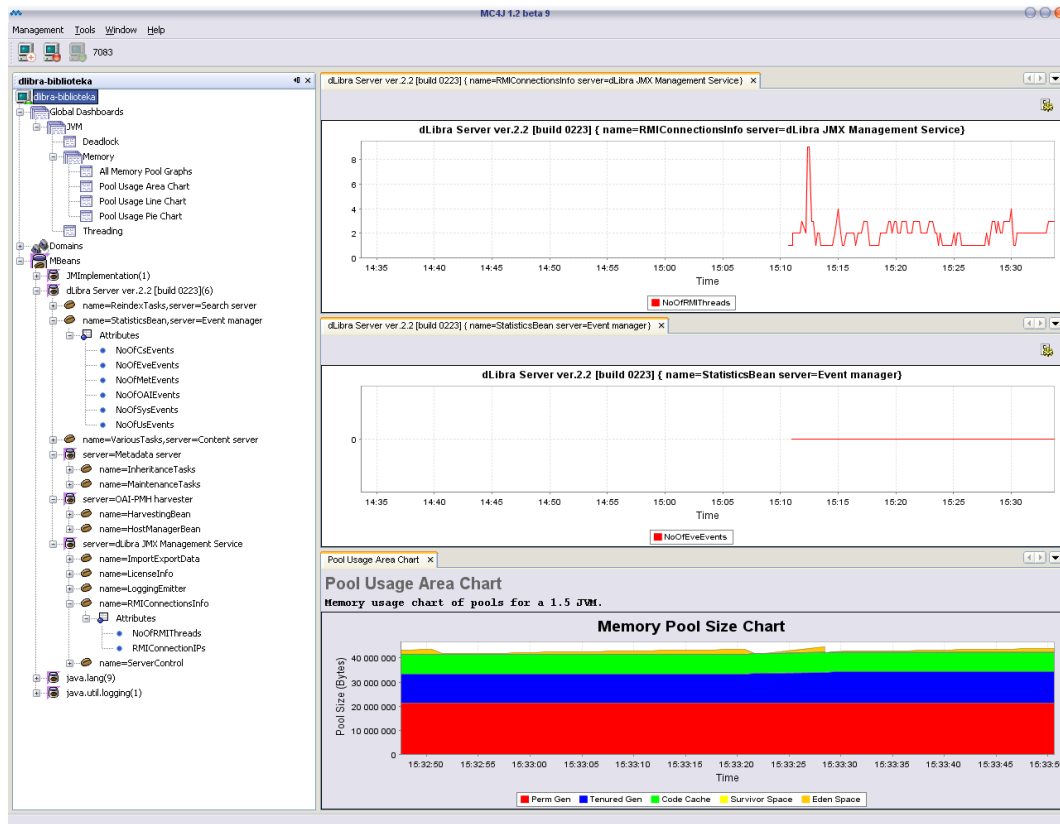
- Connection type: JSR160
- Name: dowolnie
- Initial context factory: com.sun.jndi.rmi.registry.RegistryContextFactory
- Server URL, Principle, Credentials: zgodnie z opisem poniżej

Pozostałe kroki kreatora można pominąć. Oficjalna dokumentacja MC4J znajduje się pod adresem <http://mc4j.org/confluence/display/MC4J/User+Guide> .

2.2. Zakres funkcjonalny interfejsu administracyjnego serwera

W celu podłączenia się do zdalnego interfejsu udostępnianego w technologii JMX trzeba zazwyczaj podać specjalnie sformatowany adres interfejsu oraz dane autoryzacyjne. W przypadku systemu *dLibra* adres interfejsu ma postać: **service:jmx:rmi:///jndi/rmi://[adres serwera]:[port serwera]/mx** Jako dane autoryzacyjne należy podać nazwę użytkownika i hasło dla konta 'admin' zdefiniowanego w danym serwerze *dLibra* .

Poniższy rysunek przedstawia interfejs administracyjny systemu *dLibra* obsługiwany przy pomocy programu MC4J. Po lewej stronie widać drzewo komponentów administracyjnych. Prawa strona ekranu wykorzystana jest do wyświetlenia trzech wykresów: liczby połączeń RMI do serwera *dLibry*, liczby zdarzeń w serwerze jakie pozostały do obsługi oraz zajętości pamięci.



W interfejsie administracyjnym systemu *dLibra* , poza standardowymi komponentami zarządzania udostępnianymi przez środowisko maszyny wirtualnej Java, dostępne są dedykowane komponenty zarządzania serwerem. Komponenty te podzielone są na grupy odpowiadające strukturze wewnętrznej serwera. W ramach tych grup wyróżniony jest jeszcze podział funkcjonalny. Powstaje w ten sposób hierarchiczna, drzewiasta struktura komponentów. Zgodnie ze specyfikacją Java JMX każdy z komponentów może udostępniać własności oraz metody. Poniżej opisano najistotniejsze z udostępnianych elementów. Te elementy, które pominięto są zazwyczaj elementami przydatnymi tylko w wyjątkowych sytuacjach, wymagają bardzo głębokiej znajomości struktury wewnętrznej systemu *dLibra* i nie powinny być używane. W opisie zastosowano następującą notację:

- *NazwaAtrybutu* - oznacza atrybut, którego wartość może być odczytywana i (opcjonalnie) modyfikowana.
- *nazwaMetody(parametr1,parametr2)* - oznacza metodę (polecenie), która może być wywołana. Mogą również występować metody bezparametrowe.

Ogólnie można założyć, że atrybuty mogą być wykorzystywane do monitorowania pracy serwera, a metody do wykonywania pewnych czynności konserwacyjnych, migracji danych między wersjami lub też np. odtwarzania części danych utraconych na skutek awarii.

Ostrzeżenie

Część komponentów pozwala na stosunkowo łatwe dokonanie poważnych zmian w danych przechowywanych w bibliotece cyfrowej. Część z operacji może też być bardzo czasochłonna - przykładowo operacja pełnej reindeksacji publikacji dużej biblioteki cyfrowej w zależności o wydajności serwera może trwać nawet kilka

tygodni. **Zaleca się zachowanie najwyższej ostrożności przy wykorzystywaniu interfejsu administracyjnego serwera dLibra .**

2.2.1. Content Server

ZipCache:

- *CacheTimeSpan* - Czas życia w cache najstarszego wydania (w godzinach).
- *CurrentZipCacheSize* - Aktualny rozmiar cache (w bajtach).
- *CurrentZipCacheSize* - Aktualny rozmiar cache w stosunku do rozmiaru maksymalnego (w procentach).
- *MaxZipCacheSize* - Maksymalny rozmiar cache (w bajtach).
- *NoOfCachedZipFiles* - Liczba plików zip w cache.
- *NoOfZipCacheHits* - Liczba "trafień" w cache.
- *NoOfZipCacheMisses* - Liczba "chybień" w cache.
- *OldestAccessDate* - Data dostępu do najstarszego pliku w cache.
- *ZipCacheHitRatio* - Stosunek liczby "trafień" w cache do ogólnej liczby żądań.

2.2.2. Event Manager

EventMonitoringBean:

- *NoOfEveEvents* - pokazuje liczbę zdarzeń, jakie zostały jeszcze do obsłużenia usługom serwera dLibra. Zdarzenia to np. dodanie nowej publikacji czy modyfikacja opisu, a obsługa ich może polegać np. na aktualizacji indeksów wyszukiwawczych. Liczba zdarzeń powinna przynajmniej raz na dobę (ewentualnie raz na tydzień) spadać do zera. Jeżeli tak nie jest, oznacza to, że serwer *dLibra* jest przeciążony i nie jest w stanie na bieżąco obsłużyć zadań generowanych przez użytkowników biblioteki.
- *EventProcessingDelay* - Opóźnienie w przetwarzaniu zdarzeń (w godzinach).
- *OldestEventDate* - Data najstarszego nieobsłużonego zdarzenia.

2.2.3. Metadata Server

MaintenanceTasks:

- *reloadDCPatterns()* - przeładowuje informacje na temat mapowania zestawu atrybutów zdefiniowanego w bibliotece cyfrowej na schemat DublinCore. Informacje te zawarte są w pliku `dc-pattern.properties` w podkatalogu `conf` katalogu głównego serwera.

2.2.4. OAI-PMH Harvester

HarvestingBean:

- *harvestAll()* - powoduje aktualizację informacji o publikacjach znajdujących się we wszystkich zdefiniowanych repozytoriach OAI-PMH.
- *harvestHost(hostId)* - powoduje aktualizację informacji o publikacjach znajdujących się repozytorium OAI o identyfikatorze podanym w parametrze **hostId**.
- *synchronizeAll()* - powoduje synchronizację (sprawdzenie całego repozytorium) informacji o publikacjach znajdujących się we wszystkich zdefiniowanych repozytoriach OAI-PMH.
- *synchronizeHost(hostId)* - powoduje synchronizację (sprawdzenie całego repozytorium) informacji o publikacjach znajdujących się repozytorium OAI o identyfikatorze podanym w parametrze **hostId**.

HostManagerBean:

- *Hosts* - daje dostęp do listy zdefiniowanych repozytoriów OAI-PMH.
- *addHost(repositoryUrl, defaultLanguage, hostLogoURL, mainPageURL)* - pozwala zdefiniować nowy host OAI-PMH. Parametry to:
 - **repositoryUrl** - adres interfejsu OAI-PMH repozytorium (np. <http://dlibra.psnc.pl/biblioteka/dlibra/oai-pmh-repository.xml>).
 - **defaultLanguage** - domyślny język opisu publikacji w repozytorium (np. pl).
 - **hostLogoUrl** - małe (40x16px) logo repozytorium do wyświetlania na liście wyników wyszukiwania (np. <http://dlibra.psnc.pl/biblioteka/my-oai-link.gif>).
 - **mainPageURL** - adres URL strony głównej repozytorium (np. <http://dlibra.psnc.pl/biblioteka/>).
- *updateHost(hostId, baseUrl, defaultLanguage, hasSecondsGranularity, hostName, hostLogoUrl, repositoryUrl, isDLibra)* - aktualizuje informacje o repozytorium OAI-PMH o identyfikatorze podanym w **hostId**. Pozostałe parametry mają dokładnie takie znaczenie jak w przypadku metody *addHost(...)*. Należy wypełnić wszystkie wartości parametrów - nie tylko te, których wartości mają być zmienione.
- *synchronizeHostsDatabase(url)* - synchronizuje bazę repozytoriów OAI-PMH ze zdalnym plikiem, którego adres URL podany jest w parametrze. Przykładem takiego pliku jest <http://dlibra.psnc.pl/all-dlibra-oai-hosts.csv>.
- *removeHost(hostId)* - usuwa z biblioteki cyfrowej wszystkie informacje na temat repozytorium o identyfikatorze podanym w parametrze **hostId**.

RemoteReindexTasks:

- *checkIndexesConsistency()* - sprawdza spójność lokalnych indeksów wyszukiwawczych dla wyszukiwanego rozproszonego.

- *reindexAll()* - powoduje przeindeksowanie wszystkich posiadanych informacji o zasobach zdalnych.

2.2.5. Search Server

ReindexTasks:

- *reindexAllEditions(indexMode)* - powoduje reindeksację całej zawartości biblioteki cyfrowej. Wartość parametry **indexMode** może przyjmować następujące wartości:
 - 1 - przeprowadzona ma być wyłącznie reindeksacja opisów publikacji (bez indeksów DublinCore - do wyszukiwania ogólnego),
 - 2 - przeprowadzona ma być wyłącznie reindeksacja treści publikacji,
 - 3 - 1 i 2,
 - 4 - przeprowadzona ma być wyłącznie reindeksacja opisów DublinCore - do wyszukiwania ogólnego,
 - 7 - 1, 2 i 4.
- *checkIndexesConsistency()* - sprawdza i naprawia spójność indeksów wyszukiwawczych.
- *reindexEditions(editionsIds, indexMode)* - powoduje reindeksację wydań o podanych identyfikatorach. Wartość parametru **indexMode** ma takie samo znaczenie jak w przypadku polecenia *reindexAllEditions(...)*, a jako wartość parametru **editionsIds** powinny być podane identyfikatory wydań do reindeksacji oddzielone od siebie przecinkami.

2.2.6. dLibra JMX Management Service

RMIConnectionsInfo:

- *NoOfRMIThreads* - liczba zewnętrznych połączeń do serwera *dLibra*.
- *RMIConnectionIPs* - adresy IP komputerów, z których obecnie nawiązane są połączenia do serwera *dLibra*.

ServiceWrapperManager:

- *stop()* - wyłącza serwer.
- *restart()* - restartuje serwer.