

Podręcznik administratora środowiska dLibra w wersji 2.5



**Poznańskie Centrum Superkomputerowo - Sieciowe
Projekt dLibra
Redakcja: Marcin Werla**

Podręcznik administratora środowiska *dLibra* w wersji 2.5

Marcin Werla

Copyright © 2002 - 2006 Poznańskie Centrum Superkomputerowo - Sieciowe

Spis treści

| | |
|---|----|
| Wstęp | v |
| 1. Czym jest dLibra ? | v |
| 2. Dla kogo przeznaczona jest ta instrukcja? | v |
| 3. Co zawiera ta instrukcja? | vi |
| 4. Dodatkowe źródła informacji | vi |
| I. Instalacja i uruchomienie | 1 |
| 1. Budowa systemu dLibra | 2 |
| 1.1. Architektura systemu | 2 |
| 1.2. Opis funkcjonalności komponentów | 2 |
| 1.2.1. Serwer | 3 |
| 1.2.2. Aplikacja administratora | 3 |
| 1.2.3. Aplikacja redaktora | 3 |
| 1.2.4. Aplikacja czytelnika | 4 |
| 1.3. Schemat połączeń między komponentami | 4 |
| 2. Przygotowanie do instalacji systemu dLibra | 5 |
| 2.1. Wymagane oprogramowanie | 5 |
| 2.2. Zawartość dystrybucji | 5 |
| 2.3. Parametry instalacyjne | 6 |
| 3. Proces instalacji | 7 |
| 3.1. Uruchomienie instalatora | 7 |
| 3.2. Inicjalizacja bazy danych | 8 |
| 3.3. Rozmieszczenie komponentów systemu | 8 |
| 3.3.1. Serwer | 8 |
| 3.3.2. Aplikacja administratora i redaktora | 9 |
| 3.3.3. Aplikacja czytelnika | 9 |
| 4. Uruchamianie systemu dLibra | 10 |
| 4.1. Serwer | 10 |
| 4.1.1. System licencji dLibra | 10 |
| 4.1.2. Uruchamianie serwera | 10 |
| 4.2. Aplikacja czytelnika | 12 |
| 4.3. Aplikacja administratora i redaktora | 12 |
| 5. Najczęściej spotykane problemy z instalacją systemu dLibra | 13 |
| 5.1. Instalacja serwera bazy danych Oracle i serwera systemu dLibra na tym samym komputerze | 13 |
| 5.2. Instalacja serwera Tomcat w katalogu, którego nazwa zawiera znaki spacji | 13 |
| II. Serwer dLibra | 15 |
| 6. Konfiguracja | 16 |
| 7. Panel administracyjny | 18 |
| 7.1. Budowa interfejsu administracyjnego serwera systemu dLibra | 18 |
| 7.1.1. Technologia JMX | 18 |
| 7.1.2. JConsole | 18 |
| 7.1.3. MC4J | 18 |
| 7.2. Zakres funkcjonalny interfejsu administracyjnego serwera | 19 |
| 7.2.1. Content Server | 20 |
| 7.2.2. Event Manager | 20 |
| 7.2.3. Metadata Server | 20 |
| 7.2.4. OAI-PMH Harvester | 21 |
| 7.2.5. Search Server | 22 |
| 7.2.6. dLibra JMX Management Service | 22 |
| III. Aplikacja Czytelnika | 23 |

| | |
|--|----|
| 8. Konfiguracja | 24 |
| 9. Dostosowywanie wyglądu | 25 |
| 10. Wewnętrzna struktura Aplikacji Czytelnika systemu dLibra | 26 |
| 10.1. Struktura dyskowa aplikacji | 26 |
| 10.2. Struktura logiczna aplikacji | 27 |
| 10.2.1. Strony | 27 |
| 10.2.2. Komponenty | 28 |
| 10.2.3. Akcje | 28 |
| 10.2.4. Pozostałe elementy | 28 |
| IV. Migracja z wersji 2.2 do wersji 2.5 | 30 |
| 11. Szczegółowa procedura migracji | 31 |
| 11.1. Wstęp | 31 |
| 11.2. Procedura migracji | 31 |
| 12. Migracja struktur dyskowych komponentów systemu dLibra | 32 |
| 12.1. Przygotowanie wstępnie skonfigurowanych komponentów | 32 |
| 12.2. Zmodyfikowane pliki binarne | 32 |
| 12.3. Zmodyfikowane pliki tekstowe | 33 |
| 13. Migracja struktur bazy danych | 34 |
| 14. Migracja danych | 35 |
| A. Parametry konfiguracyjne instalatora systemu dLibra | 36 |
| B. Architektura oraz możliwości skalowania systemu dLibra | 39 |
| B.1. Architektura systemu dLibra | 39 |
| B.2. Serwer systemu dLibra | 40 |
| B.3. Skalowanie systemu dLibra | 40 |
| B.3.1. Dwa serwery | 41 |
| B.3.2. Trzy serwery | 41 |

Wstęp

1. Czym jest *dLibra* ?

dLibra to pierwsze polskie środowisko służące do budowy bibliotek cyfrowych. *dLibra* wspiera cały proces publikacji treści elektronicznych - począwszy od pracy autora i redaktora, poprzez udostępnienie dzieła w Internecie, wprowadzanie poprawek i publikację kolejnych wydań, aż po opcjonalne usunięcie treści z biblioteki.



Dzięki mechanizmom umożliwiającym grupowanie dzieł zgromadzonych w bibliotece w kolekcje, redaktor biblioteki może łatwo zaprezentować czytelnikowi treść publikacji w ciekawej i przystępnej formie. Dodatkowo gromadzone dzieła opisywane są atrybutami, które można definiować w oparciu o standardy takie jak Dublin Core. Tworzenie opisów ułatwiają specjalne słowniki wartości atrybutów i synonimów tych wartości oraz funkcje importu opisów z zewnętrznych systemów obsługujących formaty Resource Description Framework (RDF) oraz MARC. Wprowadzone do systemu opisu można również eksportować (za pośrednictwem formatu RDF). Dzięki tym możliwościom, a także wyszukiwaniu w treści dzieł, czytelnik może łatwo i szybko znaleźć interesującą go publikację.

2. Dla kogo przeznaczona jest ta instrukcja?

Instrukcja ta jest przeznaczona dla osób, które chcą zainstalować i skonfigurować środowisko *dLibra*, a następnie administrować powstałą w ten sposób biblioteką cyfrową. Ma ona pomóc w przygotowaniu się do instalacji tego środowiska, w samej instalacji oraz w czynnościach związanych z dostosowaniem i uruchomieniem poszczególnych komponentów systemu po instalacji oraz w dalszej administracji i utrzymaniu systemu.

Aby poprawnie przejść przez proces instalacji i 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 odbędzie się instalacja. Powinien on również być w stanie zainstalować i skonfigurować dodatkowe oprogramowanie wymagane do działania systemu *dLibra*. Niezbędna jest również podstawowa znajomość języka angielskiego.

3. Co zawiera ta instrukcja?

Instrukcja ta opisuje czynności związane z instalacją, konfiguracją i uruchomieniem wszystkich komponentów środowiska *dLibra* . Składa się ona z następujących rozdziałów:

| | |
|---|--|
| Część I. Instalacja i uruchomienie | Zawiera ogólny opis systemu <i>dLibra</i> oraz instrukcję instalacji i uruchomienia systemu. |
| Część II. Serwer <i>dLibra</i> | Zawiera informacje związane z konfiguracją, administracją i utrzymaniem serwera systemu <i>dLibra</i> . |
| Część III. Aplikacja Czytelnika | Zawiera informacje związane z konfiguracją, dostosowaniem wyglądu i utrzymaniem Aplikacji Czytelnika systemu <i>dLibra</i> . |
| Część IV. Migracja z wersji 2.2 do wersji 2.5 | Zawiera instrukcję samodzielnej migracji z wersji 2.2 do wersji 2.5 oprogramowania <i>dLibra</i> . |
| Dodatki | Dodatki zawierają szczegółowe informacje związane z tematami poruszonymi w poszczególnych częściach tej instrukcji. |

4. 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> .

Część I. Instalacja i uruchomienie

Instrukcja ta opisuje czynności związane z instalacją, konfiguracją i uruchomieniem wszystkich komponentów środowiska *dLibra* . Składa się ona z następujących rozdziałów:

| | |
|---|--|
| Rozdział 1. <i>Budowa systemu dLibra</i> | Zawiera omówienie architektury systemu <i>dLibra</i> oraz funkcjonalności poszczególnych jego komponentów, a także schemat połączeń sieciowych pomiędzy komponentami. |
| Rozdział 2. <i>Przygotowanie do instalacji systemu dLibra</i> | Opisuje przygotowanie do instalacji systemu. Zawiera także spis oprogramowania zewnętrznych dostawców niezbędnego do uruchomienia oprogramowania <i>dLibra</i> oraz opis zawartości pakietu dystrybucyjnego systemu. |
| Rozdział 3. <i>Proces instalacji</i> | Jest poświęcony samemu procesowi instalacji. |
| Rozdział 4. <i>Uruchamianie systemu dLibra</i> | Zawiera informacje na temat uruchamiania serwera oraz aplikacji składowych systemu <i>dLibra</i> . |
| Rozdział 5. <i>Najczęściej spotykane problemy z instalacją systemu dLibra</i> | Zawiera opis problemów i ich rozwiązań, z którymi najczęściej spotykali się użytkownicy oprogramowania <i>dLibra</i> . |

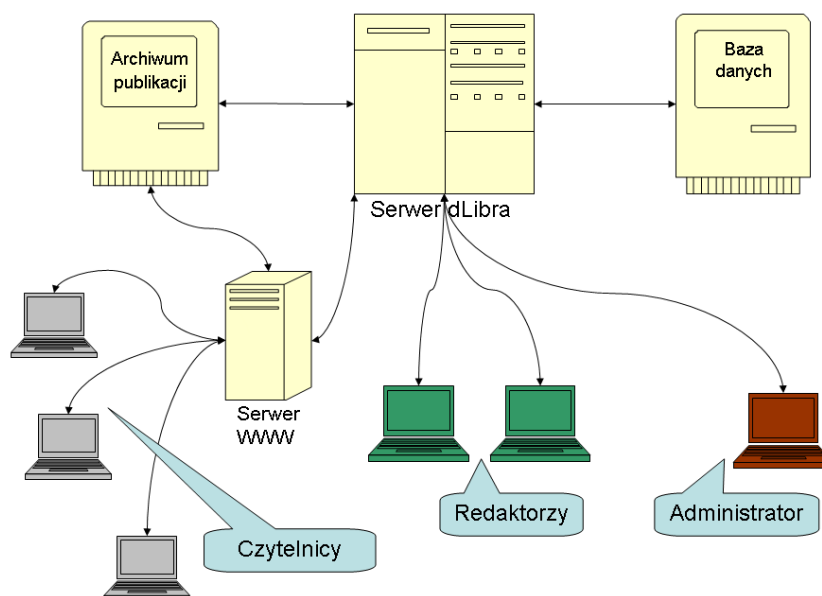
Na końcu instrukcji, jako Dodatek A. *Parametry konfiguracyjne instalatora systemu dLibra* , została zamieszczona tabela zawierająca omówienie wszystkich parametrów konfiguracyjnych instalatora systemu.

Rozdział 1. Budowa systemu *dLibra*

1.1. Architektura systemu

System *dLibra* jest systemem wielowarstwowym, wykorzystującym do połączeń pomiędzy poszczególnymi komponentami (warstwami) sieć komputerową. Poniższy rysunek schematycznie przedstawia architekturę systemu.

Rysunek 1.1. Uproszczony schemat architektury systemu *dLibra*



Taka architektura umożliwia rozmieszczenie poszczególnych komponentów systemu (czyli serwera, aplikacji do zarządzania biblioteką, aplikacji WWW, bazy danych oraz serwera archiwum publikacji) na różnych komputerach. Ważne jest jedynie, aby były one połączone między sobą siecią komputerową umożliwiającą komunikację przy użyciu protokołu TCP/IP. Oczywiście, możliwe jest zainstalowanie i uruchomienie wszystkich komponentów systemu na jednym komputerze. Szczegółowe informacje na temat architektury oraz możliwości skalowania systemu *dLibra* zawiera Dodatek B. *Architektura oraz możliwości skalowania systemu *dLibra** .

1.2. Opis funkcjonalności komponentów

System *dLibra* składa się z następujących komponentów:

- Serwer
- Aplikacja administratora
- Aplikacja redaktora
- Aplikacja czytelnika

Poniżej ogólnie opisano funkcjonalność wyżej wymienionych komponentów.

1.2.1. Serwer

Serwer to podstawowy element systemu *dLibra* - odpowiada on za działanie całej biblioteki cyfrowej. Serwer przechowuje wszystkie zebrane w bibliotece informacje, czyli zarówno opisy publikacji, jak i informacje o kontaktach użytkowników czy zdefiniowanych w danej bibliotece kolekcjach. Treść umieszczonych w bibliotece publikacji przechowywana jest na dysku komputera, na którym uruchomiono serwer. Wszystkie pozostałe dane umieszczone są w używanej przez serwer bazie danych. Obecnie wspierane są bazy danych Oracle, PostgreSQL oraz MySQL. Dodatkowo serwer ma również możliwość utrzymywania zewnętrznego archiwum treści wprowadzanych publikacji. Archiwum to może być umieszczone na dysku serwera bądź też na dowolnym innym zewnętrznym serwerze dostępnym przez protokół FTP. Użytkownicy uzyskują dostęp do pełnej funkcjonalności systemu dzięki połączeniu swojej aplikacji klienta do serwera. Autoryzacja użytkowników następuje w oparciu o wewnętrzną bazę danych serwera bądź też przy użyciu informacji dostępnych w zewnętrznych serwerach LDAP.

Szczegółowe informacje na temat serwera, możliwości jego konfiguracji oraz administracji zawiera Część II. Serwer *dLibra* .

1.2.2. Aplikacja administratora

Aplikacja administratora daje uprawnionym użytkownikom biblioteki cyfrowej możliwość zarządzania tą biblioteką. Podstawowe funkcje tej aplikacji to m.in.:

- zarządzanie użytkownikami i grupami użytkowników,
- zarządzanie katalogami i kolekcjami publikacji,
- zarządzanie atrybutami używanymi do opisu elementów biblioteki cyfrowej i słownikami wartości tych atrybutów.

Dokładny opis funkcjonalności tej aplikacji oraz jej instrukcję obsługi można znaleźć w *Podręczniku użytkownika środowiska dLibra* .

1.2.3. Aplikacja redaktora

Głównym zadaniem aplikacji redaktora jest umożliwienie zarządzania treścią zebraną w bibliotece cyfrowej. I tak, posiadający stosowne uprawnienia użytkownik aplikacji redaktora może m.in.:

- wprowadzać nowe publikacje do biblioteki,
- zarządzać istniejącymi w bibliotece publikacjami (tworzyć nowe wydania publikacji, przenosić publikacje pomiędzy katalogami, zmieniać przyporządkowanie publikacji do kolekcji, ...),
- opisywać zebrane publikacje metadanymi,
- decydować o zasadach dostępu do poszczególnych publikacji i katalogów,
- usuwać publikacje.

Dokładny opis funkcjonalności tej aplikacji oraz jej instrukcję obsługi można znaleźć w *Podręczniku użytkownika środowiska dLibra* .

1.2.4. Aplikacja czytelnika

Podstawowym sposobem korzystania z publikacji zgromadzonych w bibliotece cyfrowej opartej o system *dLibra* jest dostępna przez WWW aplikacja czytelnika. Umożliwia ona m.in.:

- przeszukiwanie zawartości biblioteki (zarówno treści publikacji jak i ich opisów),
- przeglądanie zawartości poszczególnych kolekcji,
- czytanie zgromadzonych w bibliotece publikacji,
- dostęp do statystyk popularności poszczególnych publikacji,
- dostęp do listy ostatnio dodanych do biblioteki publikacji oraz powiadomienia o ich dodaniu (przy pomocy technologii RSS Feed).

Szczegółowe informacje na temat zasad konfiguracji oraz możliwości dostosowywania wyglądu aplikacji czytelnika zawiera Część III. Aplikacja Czytelnika .

1.3. Schemat połączeń między komponentami

W celu umożliwienia pełnej współpracy pomiędzy poszczególnymi komponentami systemu *dLibra* należy tak skonfigurować połączenia sieciowe, firewalle i inne tego rodzaju urządzenia, aby możliwe były następujące połączenia:

- serwer *dLibra* -> baza danych - połączenie na porcie, na którym nasłuchuje baza danych. Domyślnie są to porty: 1521 dla Oracle, 5432 dla PostgreSQL oraz 3306 dla MySQL.
- serwer *dLibra* -> archiwum publikacji - aktywne połączenie FTP (opcjonalnie).
- serwer WWW -> serwer *dLibra* - połączenie na portach, na których nasłuchuje serwer *dLibra* . Domyślnie są to porty 10051 i 10052. Pierwszy z tych portów konfigurowany jest w pliku konfiguracyjnym serwera *dLibra* , a numer drugiego portu powstaje poprzez zwiększenie numeru portu pierwszego o jeden.
- serwer WWW -> archiwum publikacji - aktywne połączenie FTP (opcjonalnie).
- aplikacja redaktora/administratora -> serwer *dLibra* - tak samo jak w przypadku połączenia serwer WWW -> serwer *dLibra* .
- czytelnicy -> serwer WWW - połączenia na portach, na których nasłuchuje serwer WWW (domyślnie porty 80 i 443).

Dodatkowo serwer *dLibra* oraz serwer WWW mogą w pewnych okolicznościach wysyłać do użytkowników wiadomości e-mail. W tym celu niezbędne jest umożliwienie połączeń pomiędzy tymi dwoma komponentami, a serwerem poczty wychodzącej SMTP zdefiniowanym w plikach konfiguracyjnych wspomnianych komponentów.

Rozdział 2. Przygotowanie do instalacji systemu *dLibra*

2.1. Wymagane oprogramowanie

Aby zainstalować i uruchomić system *dLibra* niezbędne jest następujące oprogramowanie:

- Java Runtime Environment (JRE) - w wersji co najmniej 1.5, dostępne na stronie Sun Microsystems Inc. . Oprogramowanie to musi być zainstalowane na każdym komputerze, na którym ma być uruchomiony dowolny z elementów systemu *dLibra* , w tym również instalator systemu. Należy pamiętać o ustawieniu zmiennej systemowej **JAVA_HOME** na katalog w którym zainstalowano JRE, oraz dodaniu do zmiennej systemowej **PATH** podkatalogu `bin` katalogu, w którym została zainstalowano JRE.
- Apache Ant - w wersji co najmniej 1.6.5, dostępne na stronie Apache . Oprogramowanie to musi być zainstalowane na komputerze, na którym uruchamiany będzie instalator systemu. Należy pamiętać o ustawieniu zmiennej systemowej **ANT_HOME** na katalog w którym zainstalowano Apache Ant, oraz dodaniu do zmiennej systemowej **PATH** podkatalogu `bin` katalogu, w którym została zainstalowano Apache Ant.
- Baza danych - niezbędna, aby uruchomić serwer systemu *dLibra* . Aktualnie wspierane są następujące typy baz danych: Oracle w wersji 9.2.x, PostgreSQL w wersji 8.x oraz MySQL w wersji 5.x.

Uwaga!

Baza danych, która będzie wykorzystywana do działania systemu *dLibra* powinna być tak skonfigurowana, aby przechowywała dane w kodowaniu UTF-8 lub zgodnym.

- Serwer aplikacji zgodny ze standardem Java Servlet 2.4 - niezbędny, aby uruchomić aplikację WWW systemu *dLibra* . Przykładowym serwerem tego typu jest Apache Tomcat w wersji 5.x (zalecana jest najnowsza wersja z serii 5). Szczegółowe informacje dostępne są na stronie projektu .

2.2. Zawartość dystrybucji

Struktura katalogów dystrybucji systemu *dLibra* opisano poniżej.

- `binaries` - zawiera wersje wykonywalne poszczególnych komponentów systemu. Poszczególne podkatalogi zawierają:
 - `dlibra-app` - aplikacja administratora i redaktora w dwóch wersjach: w wersji przeznaczonej do bezpośredniego uruchamiania (podkatalog `standalone`) oraz w wersji przygotowanej do uruchamiania przy wykorzystaniu protokołu JNLP i technologii Java WebStart (podkatalog `jnlp`).
 - `dlibra-server` - serwer systemu *dLibra* .

- **dlibra-webapp** - aplikacja czytelnika (WWW).
- **manuals** - polska i angielska dokumentacja system w formacie PDF. Poszczególne podkatalogi zawierają:
 - **dlibra-app** - *Podręcznik użytkownika środowiska dLibra* opisujący zasady używania aplikacji edytora i aplikacji administratora.
 - **dlibra-server** - *Instrukcja instalacji systemu dLibra* , czyli ten dokument.
- **migrate-sql** - pliki zawierające polecenia SQL wykorzystywane do migracji struktur danych między wersjami systemu dLibra - patrz Rozdział 4. *Uruchamianie systemu dLibra* .
- **sql** - pliki zawierające polecenia SQL wykorzystywane do zainicjowania bazy danych.

Dodatkowo w głównym katalogu pakietu instalacyjnego znajdują się następujące pliki:

- **build.xml** - skrypt instalatora zgodny z interpreterem Apache Ant.
- **sample.properties** - przykładowa konfiguracja instalatora.

2.3. Parametry instalacyjne

Przed przystąpieniem do instalacji należy przygotować plik tekstowy zawierający parametry konfiguracji instalacji systemu *dLibra* . Plik ten składa się z kolejnych wpisów w postaci: **<nazwa> = <wartosc>** . Każdy wpis powinien znajdować się w osobnej linii. Przykładowe wartości poszczególnych parametrów zawiera plik **sample.properties** znajdujący się w głównym katalogu pakietu dystrybucyjnego. Znaczenie poszczególnych parametrów zawiera Dodatek A. *Parametry konfiguracyjne instalatora systemu dLibra* .

Notatka

Warto zachować plik konfiguracyjny wykorzystany w procesie instalacji. Może on być później wykorzystany przy migracji do nowszych wersji oprogramowania *dLibra* .

Rozdział 3. Proces instalacji

Po zainstalowaniu niezbędnego oprogramowania zewnętrznych dostawców (patrz punkt Wymagane oprogramowanie), można rozpocząć proces instalacji systemu *dLibra* . Składa się on z opisanych poniżej trzech kroków. Pierwszy z nich, to uruchomienie instalatora w celu zebrania odpowiednich informacji i przygotowania odpowiednio skonfigurowanych pakietów oprogramowania. Drugi to zainicjowanie bazy danych przy pomocy skryptów wygenerowanych przez instalator. Trzeci krok polega na odpowiednim rozmieszczeniu wygenerowanych przez instalator pakietów oprogramowania.

3.1. Uruchomienie instalatora

Instalator uruchamia się poprzez uruchomienie interpretera skryptów Apache Ant a katalog głównym pakietu instalacyjnego. Interpreter automatycznie wczyta znajdujący się tam skrypt instalatora. Po uruchomieniu tego skryptu użytkownik będzie musiał odpowiedzieć na przedstawione poniżej pytania:

```
Do you accept dLibra software license?
```

Odpowiedź **yes** na powyższe pytanie oznacza zaakceptowanie licencji oprogramowania *dLibra* i umożliwia rozpoczęcie procesu instalacji. Zasady licencjonowania dostępne są na stronie projektu *dLibra* .

```
Please enter configuration file name:?
```

Odpowiedzią na to pytanie powinna być nazwa pliku (wraz z pełną ścieżką), który zawiera parametry instalacyjne systemu *dLibra* (patrz punkt Przygotowanie do instalacji systemu *dLibra*).

```
Please enter target directory for dLibra framework applications:
```

Odpowiedź na powyższe pytanie określa katalog docelowy, do którego zostaną skopiowane skonfigurowane komponenty systemu *dLibra*.

Po podaniu katalogu instalator rozpocznie faktyczny proces instalacji, w trakcie którego nastąpi stworzenie plików konfiguracyjnych poszczególnych komponentów systemu oraz skopiowanie skonfigurowanych komponentów do katalogu docelowego. Po zakończeniu instalacji katalog docelowy będzie miał następującą strukturę:

- Podkatalog `dlibra-app-standalone-[numer.wersji]` - zawiera "Aplikację Redaktora i Administratora" w postaci uruchamialnej. Może ona być wykorzystana przez osobę instalującą oprogramowanie *dLibra* do sprawdzenia poprawności instalacji serwera systemu przed uruchomieniem aplikacji czytelnika.
- Podkatalog `dlibra-server-[numer.wersji]` - zawiera przygotowany do uruchomienia serwer *dLibra* .

- Podkatalog `dlibra-webapp-[numer.wersji]` oraz plik `dlibra.war` - gotowa do użycia aplikacja czytelnika systemu *dLibra* z wbudowaną aplikacją redaktora/administratora. Katalog i plik `war` mają dokładnie taką samą zawartość. Należy zdecydować się na wykorzystanie jednego z nich w zależności od preferencji konfiguracji wybranego serwera aplikacji.
- Podkatalog `sql` - zawiera pliki z poleceniami SQL, które mają być wykorzystane do inicjalizacji bazy danych (patrz niżej). Pliki zapisane są w kodowaniu UTF-8.

3.2. Inicjalizacja bazy danych

Uwaga!

Baza danych, która będzie wykorzystywana do działania systemu *dLibra* powinna być tak skonfigurowana, aby przechowywała dane w kodowaniu UTF-8 lub zgodnym.

Po poprawnym zakończeniu działania instalatora należy zainicjować bazę danych. W tym celu należy wykorzystać pliki z poleceniami SQL znajdujące się w podkatalogu `sql` katalogu docelowego wybranego podczas procesu instalacji. W katalogu tym znajdują się grupa plików do tworzenia struktury bazy danych dla poszczególnych usług (`dlibra-server-[id.usługi]-[numer.wersji]-[rodzaj.bazy]-schema-create.sql`), grupa plików do usuwania bazy danych dla poszczególnych usług (`dlibra-server-[id.usługi]-[numer.wersji]-[rodzaj.bazy]-schema-drop.sql`) oraz plik `dlibra-server-[numer.wersji]-init.sql` który pozwala zainicjować uprzednio stworzone struktury bazy danych. Pliki te są zapisane w kodowaniu UTF-8 i niezbędne jest, aby dane w nich zawarte zostały wprowadzone do bazy danych tak, aby nie nastąpiło przekłamanie związane ze zmianą kodowania znaków. Przykładowe oprogramowanie umożliwiające wprowadzenie takich danych do wielu różnych rodzajów baz danych to bezpłatny *Squirrel SQL* bazujący na języku Java i protokole JDBC.

W celu przygotowania bazy danych dla serwera systemu *dLibra* , należy najpierw stworzyć struktury bazy danych dla poszczególnych usług (kolejność usług nie jest istotna), a następnie wypełnić je danymi początkowymi. Sposób wykonania opisanych wyżej poleceń zależał będzie od użytej bazy danych oraz od oprogramowania zainstalowanego na komputerze, z którego przeprowadzana będzie ta operacja.

3.3. Rozmieszczenie komponentów systemu

Po poprawnym zakończeniu działania instalatora i zainicjalizowaniu bazy danych należy przenieść skonfigurowane komponenty systemu we właściwe lokalizacje. Poniżej opisano tę czynność dla każdego z elementów systemu.

3.3.1. Serwer

W wyniku instalacji w docelowym katalogu powstaje podkatalog o nazwie `dlibra-server-[numer.wersji]` , zwanym dalej katalogiem głównym serwera. Zawiera on przygotowany do uruchomienia serwer systemu *dLibra* . Serwer może być uruchomiony bezpośrednio w katalogu, w którym został umieszczony w procesie instalacji. Można też przenieść ten katalog w dowolne inne miejsce na dysku. Dopuszczalna jest zmiana nazwy katalogu z `dlibra-server-[numer.wersji]` na dowolną inną.

3.3.2. Aplikacja administratora i redaktora

Znajduje się ona w pliku `dlibra.war` i katalogu `dlibra-webapp-[numer.wersji]`, które są w docelowym katalogu instalacji. Uruchomienie jej opisane jest w następnym punkcie dotyczącym aplikacji czytelnika.

3.3.3. Aplikacja czytelnika

Jeżeli używanym kontenerem serwletów jest zalecany Apache Tomcat, to plik `dlibra.war` z aplikacją czytelnika należy przenieść z katalogu docelowego instalatora do podkatalogu `webapps` w katalogu, w którym został zainstalowany Tomcat. Można też tak skonfigurować Tomcata, aby wykorzystywał aplikację czytelnika umieszczoną w katalogu `dlibra-webapp-[numer.wersji]`.

W przypadku używania innego kontenera serwletów, sposób osadzenia aplikacji czytelnika w kontenerze należy sprawdzić w dokumentacji tego kontenera.

Rozdział 4. Uruchamianie systemu *dLibra*

Kolejność uruchamiania

Pierwszym uruchomionym komponentem systemu powinna być baza danych, a drugim serwer systemu *dLibra*. Pozostałe komponenty można włączać w dowolnej kolejności, gdyż korzystają one tylko z serwera i są między sobą niezależne.

4.1. Serwer

4.1.1. System licencji *dLibra*

Do uruchomienia serwera systemu *dLibra* niezbędne jest posiadanie stosownego pliku licencji. Zawiera on podstawowe informacje na temat licencjobiorcy, adres IP komputera, na którym będzie uruchamiany serwer *dLibra* i informacje o samej licencji: rodzaj licencji, datę wygenerowania licencji oraz dla licencji demo również informacje o nałożonych ograniczeniach. W celu uzyskania takiego pliku prosimy o kontakt pod adresem dlibra@man.poznan.pl. Otrzymany plik należy umieścić w katalogu domowym użytkownika, który będzie uruchamiał serwer *dLibra*.

4.1.2. Uruchamianie serwera

Serwer systemu *dLibra* może działać w dwóch trybach: w trybie usługi lub w trybie wyświetlania komunikatów na konsolę.

W trakcie normalnej pracy serwer powinien być uruchomiony w trybie usługi. Nie wymaga on wtedy, aby jakikolwiek użytkownik był zalogowany na komputerze, na którym uruchomiono serwer, a wszystkie informacje o pracy serwera trafiają do plików znajdujących się w podkatalogu **logs** katalogu głównego serwera. Uruchamianie serwera jako usługi w poszczególnych systemach operacyjnych opisano poniżej.

- UNIX/Linux - użytkownik może uruchomić skrypt `run-server.sh` znajdujący się w katalogu głównym serwera z następującymi parametrami:
 - **start** - uruchamia serwer *dLibra* w trybie usługi,
 - **stop** - zatrzymuje działający serwer *dLibra*,
 - **restart** - restartuje działający serwer *dLibra*,
 - **console** - uruchamia serwer *dLibra* w trybie konsolowym (patrz niżej).
- MS Windows - użytkownik może uruchomić skrypt `run-server.bat` znajdujący się w katalogu głównym serwera z następującymi parametrami:
 - **start** - uruchamia serwer *dLibra* w trybie usługi,
 - **stop** - zatrzymuje działający serwer *dLibra*,

4.2. Aplikacja czytelnika

Uruchomienie aplikacji czytelnika sprowadza się do uruchomienia Tomcata. Po jego uruchomieniu aplikacja WWW powinna być dostępna pod adresem `http://localhost:8080/dlibra/` przy założeniu domyślnej konfiguracji Tomcata i standardowej nazwy katalogu aplikacji WWW systemu *dLibra* .

Dodatkowo pod adresem `http://localhost:8080/dlibra/dlibra/admin` powinna być dostępna część administracyjna aplikacji WWW.

4.3. Aplikacja administratora i redaktora

Aplikację administratora i redaktora można uruchomić na dwa sposoby. Pierwszym sposobem jest wpisanie w przeglądarce WWW adresu `<adres aplikacji czytelnika>/jnlp/dlibra-app.jnlp` czyli dla domyślnej konfiguracji jest to `http://localhost:8080/dlibra/jnlp` .

Drugim sposobem uruchomienia aplikacji (bez wykorzystania JNLP) jest uruchomienie skryptu `run.sh` lub `run.bat` odpowiednio dla systemu UNIX/Linux i systemu z rodziny MS Windows, znajdującego się w katalogu głównym wersji aplikacji przeznaczonej do osobnego uruchamiania.

Rozdział 5. Najczęściej spotykane problemy z instalacją systemu dLibra

5.1. Instalacja serwera bazy danych Oracle i serwera systemu *dLibra* na tym samym komputerze

Podczas instalacji serwera bazy danych Oracle w domyślnej konfiguracji, w wersji 9.0, instalowana jest wirtualna maszyna Javy w wersji 1.3.1. Katalog `bin` tej instalacji Javy jest dodawany do zmiennej systemowej `PATH`. Z tego powodu próba uruchomienia serwera systemu *dLibra* kończy się niepowodzeniem - serwer powinien być uruchamiany w środowisku zgodnym z Java 1.4. Aby zapobiec takiej sytuacji, należy po instalacji serwera Oracle i maszyny wirtualnej Javy w wersji zgodnej z 1.4, sprawdzić czy zmienna systemowa `PATH` wskazuje na katalog `bin` właściwej wersji Javy.

Podczas instalacji serwera bazy danych Oracle domyślnie instalowana jest również XML-owa baza danych (XDB). Domyślnie dostęp do niej odbywa się poprzez port 8080. Port ten jest również domyślnym portem dla wielu kontenerów serwetów - w tym proponowanego do wykorzystania w systemie *dLibra* kontenera Tomcat. Z tego powodu nie jest możliwe równoczesne działanie bazy XDB i kontenera Tomcat na jednym komputerze, jeżeli wykorzystywane są ich domyślne konfiguracje. Aby zmienić tą sytuację można zmodyfikować konfigurację Tomcata lub XDB.

Aby zmodyfikować port, którego używa Tomcat należy zmienić wartości 8080 występujące w pliku `server.xml` w podkatalogu `conf` katalogu, w którym zainstalowano Tomcata. Można je zmienić na przykład na wartość 80.

Natomiast aby zmodyfikować port używany przez XDB należy podłączyć się do bazy danych jako użytkownik `SYSDBA` i wydać następujące zapytania:

```
SQL> call dbms_xdb.cfg_update(updateXML(dbms_xdb.cfg_get(),
'/xdbconfig/sysconfig/protocolconfig/httpconfig/http-port/text()',8081))
/
SQL> commit;
SQL> exec dbms_xdb.cfg_refresh;
```

5.2. Instalacja serwera Tomcat w katalogu, którego nazwa zawiera znaki spacji

Przy próbie uruchomienia Aplikacji Czytelnika może zdarzyć się aplikacja nie będzie działała poprawnie, a w logach Tomcata znajdą się komunikaty o błędach zbliżone do poniższych:

Najczęściej spotykane problemy z instalacją systemu dLibra

```
...
java.rmi.ServerException: RemoteException occurred in server thread; nested exception is:
  java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is:
    java.net.MalformedURLException: no protocol: Files/Apache
...
```

Przyczyną takiej sytuacji jest błąd w JDK, który uniemożliwia wykorzystanie technologii RMI w aplikacjach uruchamianych w Tomcacie, jeżeli jest on zainstalowany w katalogu, który zawiera spację (np. C:\Program Files\Apache Group\Jakarta Tomcat 5.0). Niestety obecnie jedyną znaną wyjściem z tej sytuacji jest ponowna instalacja Tomcata w katalogu, którego ścieżka nie zawiera znaków spacji. Szczegółowe informacje można znaleźć pod adresem: http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4273532

Część II. Serwer dLibra

Ta część dotyczy serwera systemu *dLibra* . Składa się ona z następujących rozdziałów:

| | |
|--|---|
| Rozdział 6. <i>Konfiguracja</i> | Zawiera omówienie poszczególnych plików konfiguracyjnych serwera. |
| Rozdział 7. <i>Panel administracyjny</i> | Opisuje panel administracyjny serwera realizowany w oparciu o technologię Java JMX. |

Rozdział 6. Konfiguracja

Uwaga!

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

W podkatalogu **conf** katalogu głównego serwera znajdują się m.in. następujące pliki konfiguracyjne:

- `database.properties` - zawiera nazwę użytkownika, hasło oraz adres bazy danych wykorzystywanej przez serwer *dLibra* oraz parametry definiujące rodzaj wykorzystywanej bazy danych.
- `server.xml` - główny plik konfiguracyjny serwera. Zawiera on większość parametrów omówionych w załączniku A.
- `wrapper.conf` - plik konfiguracyjny 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`).
- `quartz.properties` - plik konfigurujący mechanizm zadań okresowych.
- `log4j.properties` - plik konfigurujący mechanizm logowania przebiegu pracy serwera.
- pliki `??/jobs.xml` - pliki konfigurujące zadania okresowe na poziomie poszczególnych usług serwera *dLibra*.
- `cs/zip_readme.txt` - szablon informacji tekstowej dołączanej do archiwów z treścią publikacji.
- `ms/dc-pattern.properties` - plik określający zasady transformacji metadanych do schematu DublinCore.
- `se/diacritic.properties` - plik określający zasady transformacji znaków diakrytycznych na potrzeby indeksowania i wyszukiwania.
- `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.1=host
ldap.server.base.dn.1=baseDN
ldap.admin.dn.1=adminDN
ldap.admin.password.1=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.1=prefix
domain.name.1.1=domain
...
domain.name.1.m=domain
...
domain.search.prefix.n=prefix
domain.name.n.1=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 referrali 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 7. Panel administracyjny

7.1. Budowa interfejsu administracyjnego serwera systemu dLibra

7.1.1. 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 .

7.1.2. 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.

7.1.3. 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

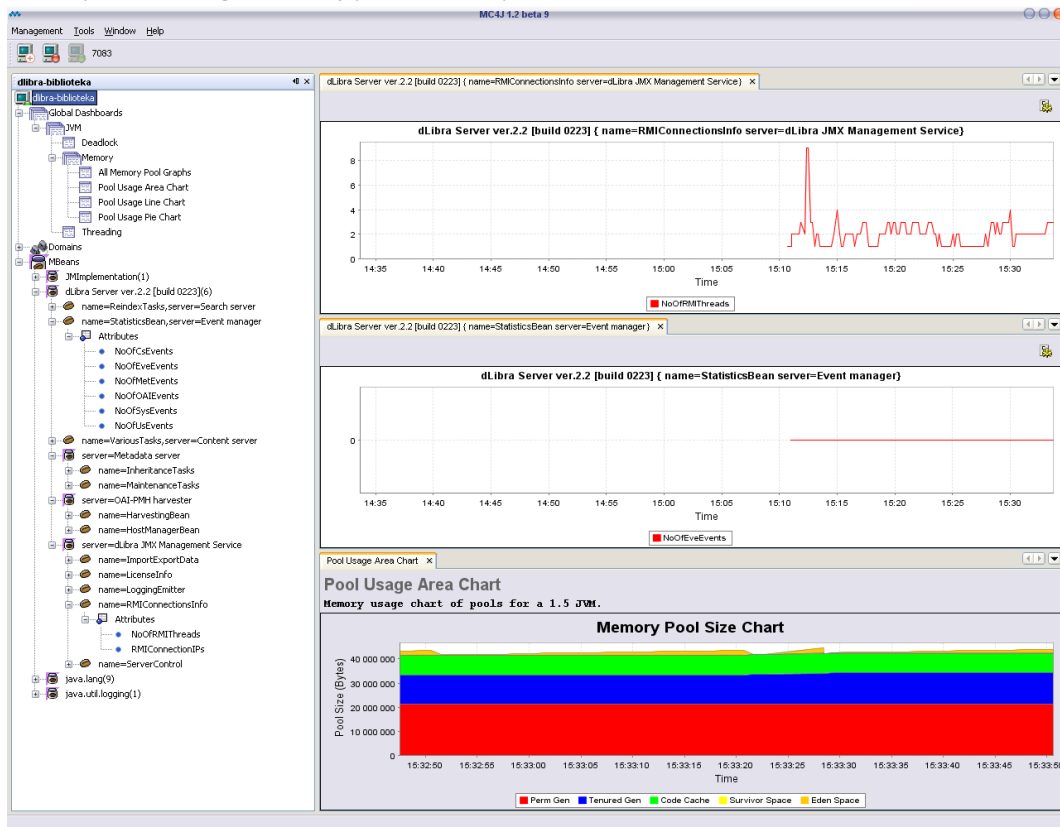
- Initial context factory: com.sun.jndi.rmi.registry.RegistryContextFactory
- Server URL, Principle, Credentials: zgodnie z opisem poniżej

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

7.2. 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 .**

7.2.1. Content Server

Various Tasks:

- *recreateZIPArchive()* - odtwarza archiwum ZIP publikacji przechowywanych w bibliotece cyfrowej. Przydatne, gdy zdecydowano się na włączenie funkcji archiwum ZIP dopiero po pewnym czasie od utworzenia biblioteki cyfrowej.

7.2.2. Event Manager

StatisticsBean:

- *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.

7.2.3. 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.

7.2.4. 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** .

HostManagerBean:

- *Hosts* - daje dostęp do listy zdefiniowanych repozytoriów OAI-PMH.
- *addHost(baseUrl, defaultLanguage, hasSecondsGranularity, hostName, hostLogoUrl, repositoryUrl, isDLibra)* - pozwala zdefiniować nowy host OAI-PMH. Parametry to:
 - **baseUrl** - adres URL strony głównej repozytorium (np. <http://dlibra.psnc.pl/biblioteka/>).
 - **defaultLanguage** - domyślny język opisu publikacji w repozytorium (np. pl).
 - **hasSecondsGranularity** - czy repozytorium przechowuje informacje o czasie modyfikacji rekordów z dokładnością do sekund.
 - **hostName** - nazwa repozytorium (np. Biblioteka Cyfrowa Projektu dLibra).
 - **hostLogoUrl** - małe (40x16px) logo repozytorium do wyświetlania na liście wyników wyszukiwania (np. <http://dlibra.psnc.pl/biblioteka/my-oai-link.gif>).
 - **repositoryUrl** - adres interfejsu OAI-PMH repozytorium (np. <http://dlibra.psnc.pl/biblioteka/dlibra/oai-pmh-repository.xml>).
 - **isDLibra** - czy repozytorium oparte jest na oprogramowaniu dLibra).
- *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** .

7.2.5. 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.
- *mergeVersionIndicies()* - powoduje połączenie dobowego i pełnego indeksu treści (standardowo odbywa się to automatycznie o 6.00 rano).
- *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.

7.2.6. dLibra JMX Management Service

LicenseInfo:

- *License* - informacje zawarte w pliku licencji.

RMICconnectionsInfo:

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

ServiceWrapperManager:

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

Część III. Aplikacja Czytelnika

Aplikacja czytelnika systemu dLibra przeznaczona jest od przeglądania, przeszukiwania i wyświetlania/pobierania treści gromadzonej w bibliotece cyfrowej. Dodatkowo umożliwia ona dostęp do biblioteki cyfrowej przy pomocy protokołów opartych na protokole HTTP takich jak RSS czy OAI-PMH. Aplikacja czytelnika zrealizowana jest w technologii Java Servlet . Do generowania stron WWW wykorzystywany jest dodatkowo system szablonów o nazwie Velocity .

Ta część opisuje strukturę i konfigurację Aplikacji Czytelnika. Składa się ona z następujących rozdziałów:

| | |
|--|--|
| Rozdział 8. <i>Konfiguracja</i> | Zawiera omówienie plików konfiguracyjnych Aplikacji Czytelnika. |
| R o z d z i a ł 9 . <i>D o s t o s o w y w a n i e</i> <i>wyglądu</i> | Opisuje możliwości dostosowania wyglądu stron generowanych przez Aplikacji Czytelnika do potrzeb danej instalacji. |
| Rozdział 10. <i>Wewnętrzna</i> <i>struktura Aplikacji</i> <i>Czytelnika systemu</i> <i>dLibra</i> | Przedstawia wewnętrzną strukturę aplikacji czytelnika. |

Rozdział 8. Konfiguracja

Uwaga!

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

Poniżej podano ścieżki ważniejszych plików konfiguracyjnych aplikacji czytelnika, wraz z opisem tych plików. Wszystkie podane ścieżki są ścieżkami względnymi do katalogu głównego tej aplikacji.

Podstawowe ustawienia

- WEB-INF/actions.xml - zawiera szablon wiadomości, która jest przesyłana bo obsługi technicznej biblioteki, gdy czytelnik skorzysta z formularza kontakt na stronach WWW oraz adres e-mail obsługi technicznej biblioteki.
- WEB-INF/ignored_ips.txt - zawiera listę adresów IP (każdy adres w nowej linii), które mają być ignorowane przy zliczaniu statystyk biblioteki cyfrowej.
- WEB-INF/web.xml - główny plik konfiguracyjny, zawiera większość parametrów podanych podczas procesu instalacji.

Komunikaty, etykiety, etc.

Notatka

Zaleca się, aby nie modyfikować poniżej wymienionych plików, a jedynie tworzyć nowe pliki o takiej samej strukturze, lecz zmienionej nazwie i tam umieszczać poprawianą/dodawaną zawartość. Nazwa nowych plików powinna być następująca: `<stara_nazwa>-custom.xml` dla plików XML oraz `<stara_nazwa>-custom_<kod_jezyka>.properties` dla plików *.properties. Zawartość tak nazwanych plików będzie automatycznie nadpisywała zawartość plików oryginalnych, a dużo prostsze będzie uaktualnianie aplikacji czytelnika do nowych wersji.

- WEB-INF/res/doctype.xml - zawiera informacje z ostrzeżeniami wyświetlanymi przy dostępie do publikacji zapisanych w określonym formacie.
- WEB-INF/res/error.xml - zawiera informacje z komunikatami o błędach.
- WEB-INF/res/information.xml - zawiera dłuższe statyczne teksty wyświetlane na stronach WWW biblioteki.
- WEB-INF/res/webapp_<kod_jezyka>.properites - zawiera etykiety przycisków, odnośników itp. wyświetlane na stronach WWW.
- WEB-INF/res/webapp_custom<kod_jezyka>.properites - zawiera nazwę biblioteki cyfrowej, dane adresowe instytucji odpowiadającej za bibliotekę itp.

Rozdział 9. Dostosowywanie wyglądu

Aplikacja czytelnika systemu *dLibra* ma możliwość dostosowywania wyglądu poprzez system schematów i szablonów. Możliwe jest definiowanie szablonów wyglądu stron WWW, przy czym każdy z szablonów może mieć dodatkowo kilka schematów np. kolorystycznych.

W pakiecie dystrybucyjnym znajduje się schemat o nazwie 'dlibra' wraz z pięcioma wariantami kolorystycznymi:

- brązowym (theme1),
- zielonym (theme2),
- fioletowym (theme3),
- szarym (theme4),
- turkusowym (theme5).

Nazwę schematu i szablonu definiuje się w pliku `WEB-INF/web.xml` edytując wartość parametru `view.style` tak, aby odpowiadał on wzorcowi `<nazwa-szablonu>:<nazwa-schematu>`. Pliki szablonu znajdują się w katalogu `WEB-INF/templates/<nazwa-szablonu>`, a pliki schematów domyślnie znajdują się w katalogu `style/<nazwa-szablonu>/<nazwa-schematu>`.

Dla domyślnego szablonu 'dlibra' przewidziano możliwość dostosowania następujących elementów graficznych (wszystkie nazwy plików odnoszą się do plików w katalogu `style/dlibra/<nazwa-schematu>`):

- logo biblioteki - plik `header/head_logo_<kod-jezyka>.jpg`, wzorzec w pliku `header/head_logo_0.jpg`
- nazwa biblioteki (duża) - plik `header/head_title_<kod-jezyka>.jpg`, wzorzec w pliku `header/head_title_0.jpg`
- nazwa biblioteki (mała) - plik `header/head_title_mini_<kod-jezyka>.jpg`, wzorzec w pliku `header/head_title_mini_0.jpg`
- logo instytucji odpowiedzialnej za bibliotekę cyfrową - plik `footer/logo_partnera.jpg`

Rozdział 10. Wewnętrzna struktura Aplikacji Czytelnika systemu dLibra

10.1. Struktura dyskowa aplikacji

W katalogu głównym Aplikacji Czytelnika znajdują się między innymi następujące pliki:

- index.html, dlibra.html - początkowe pliki nawigacyjne Aplikacji Czytelnika.
- index-loading.html - plik wyświetlany w trakcie uruchamiania Aplikacji Czytelnika.
- index-off.html - przykładowy plik, który może być wykorzystany w trakcie przerwy technicznej w działaniu Aplikacji Czytelnika.
- *.rss, *.rdf - generowane okresowo pliki kanałów RSS.
- pubViewer.jar - applet do przeglądania zabezpieczonych publikacji HTML.

Dodatkowo w katalogu tym znajdują się następujące podkatalogi:

- exlibris - opcjonalny katalog zawierający ekslibrisy poszczególnych instytucji umieszczających swoje publikacje w bibliotece cyfrowe. Ekslibrisy te mogą być automatycznie wyświetlane na stronie z opisem bibliograficznym publikacji. Mechanizm ten jest szczególnie przydatny w regionalnych bibliotekach cyfrowych.
- jnlp - katalog z aplikacją redaktora uruchamianą przy pomocy technologii Java WebStart .
- stats - automatycznie generowany katalog ze statystykami biblioteki cyfrowej.
- style - katalog z obrazkami, plikami styli CSS itp. wykorzystywanymi w aplikacji czytelnika. Zawiera on następujące podkatalogi:
 - admin - pliki wykorzystywane w panelu administracyjnym WWW.
 - common - pliki graficzne itp. niezależne od aktualnie wybranego stylu prezentacji Aplikacji Czytelnika.
 - dlibra - pliki graficzne wykorzystywane w domyślnym szablonie prezentacji Aplikacji Czytelnika.
- WEB-INF - katalog niedostępny przez przeglądarkę WWW, zawierający pliki realizujące funkcjonalność Aplikacji Czytelnika. Zawiera on następujące katalogi:
 - classes, lib - katalogi z plikami binarnymi realizującymi funkcjonalność Aplikacji Czytelnika.
 - res - katalog z plikami tekstowymi zawierającymi statyczne informacje tekstowe wykorzystywane w Aplikacji Czytelnika.
 - templates - katalog zawierający szablony wykorzystywane w Aplikacji Czytelnika.

oraz pliki:

- actions.xml, components.xml - pliki konfiguracyjne komponentów i akcji (patrz niżej).
- diacritic.properties - plik konfigurujący konwersję znaków diakrytycznych.
- ignored_agents.txt, ignored_ips.txt - pliki z wyrażeniami regularnymi opisującymi wartości pola user-agent oraz adresy IP. Wystąpienie tych wartości w żądaniach HTTP będzie powodowało wyłączenie tych żądań ze statystykach Aplikacji Czytelnika.
- pages.xml - plik konfiguracyjny stron (patrz niżej)
- periodic.xml - plik konfiguracyjny zadań okresowych wykonywanych w Aplikacji Czytelnika.
- templates.properties - plik konfiguracyjny z dodatkowymi ustawieniami szablonów Aplikacji Czytelnika.
- velocity.properties - plik konfiguracyjny mechanizmu generowania szablonów Aplikacji Czytelnika.
- web.xml - główny plik konfiguracyjny Aplikacji Czytelnika.

10.2. Struktura logiczna aplikacji

Główną składową Aplikacji Czytelnika są strony (np. strona główna, strona opisu kolekcji, strona opisu publikacji, strona wyników wyszukiwania itp.). Każda ze stron składa się z pewnych komponentów (np. komponent drzewa kolekcji, komponent formularza wyszukiwania itp.). Dodatkowym elementem są akcje - pewne czynności, które czytelnik może wykonać (np. wyszukiwanie, przesłanie formularza kontaktu, zalogowanie się itp.). Te trzy rodzaje elementów opisano szczegółowo poniżej. Statystyki czasu działania poszczególnych stron, komponentów i akcji dostępne są pod adresem /dlibra/admin-engine-stats.

10.2.1. Strony

Mechanizm stron odpowiedzialny jest za wyświetlanie czytelnikowi stron biblioteki cyfrowej po wysłaniu przez czytelnika żądania HTTP. Mechanizm ten odpowiada na wszystkie żądania skierowane pod adres <http://bibliotekacyfrowa/dlibra/>. Każda ze stron jest jednoznacznie identyfikowana przez ciąg znaków, który wystąpi po dlibra/ a przed znakiem '?'. Przykładowe identyfikatory stron to:

- metadataquery - wyszukiwanie zaawansowane,
- editionindex - lista publikacji w kolekcji,
- latest - ostatnio dodane publikacje,
- collectionstats - statystyki kolekcji,
- collectiondescription - opis kolekcji i wyszukiwanie ogólne.

Wszystkie strony zdefiniowane są w pliku /WEB-INF/pages.xml. W pliku tym określone jest, jaki komponenty wchodzi w skład strony, jakie akcje mogą być wykonane przy generowaniu strony, jakie błędy są dozwolone przy generowaniu strony, jaki jest poziom uprawnień niezbędny do wyświetlenia strony oraz czy strona ma być dostępna przez połączenie standardowe czy szyfrowane (https).

10.2.2. Komponenty

Komponenty odpowiadają za generowanie informacji wykorzystywanych w poszczególnych fragmentach strony. Dla przykładu komponenty występujące na stronie głównej Aplikacji Czytelnika systemu dLibra to:

- komponent informacji ogólnych o bibliotece cyfrowej,
- komponent drzewa kolekcji,
- komponent ostatnio dodanych publikacji,
- komponent wyszukiwania ogólnego,
- komponent wiadomości,
- komponent wystaw.

Każdy z komponentów może posiadać swoje opcje konfiguracyjne. Zapisane są one w pliku /WEB-INF/components.xml.

10.2.3. Akcje

W pewnych okolicznościach przed wyświetleniem konkretnej strony możliwe jest wykonanie pewnych akcji. Dla przykładu:

- przed wyświetleniem strony z wynikami wyszukiwania może być wykonywana akcja wyszukania publikacji,
- przed wyświetleniem strony z informacjami o profilu czytelnika może być wykonana akcja logowania,
- przed wyświetleniem strony z potwierdzeniem przesłania formularza kontaktu może być wykonana akcja przesłania tego formularza,
- przed wyświetleniem strony wyszukiwania zaawansowanego może być wykonana akcja wyczyszczenia tego zapytania lub akcja dodania pola do zapytania.

Każda z akcji może posiadać swoje opcje konfiguracyjne. Zapisane są one w pliku /WEB-INF/actions.xml.

10.2.4. Pozostałe elementy

Poza mechanizmem stron, komponentów i akcji umieszczonym pod adresem /dlibra dostępne są jeszcze cztery adresy głównego poziomu:

- /Content - odpowiedzialny za dostęp do treści publikacji. Wywołanie /Content/2952/ powoduje przejście do treści głównego pliku wydania o identyfikatorze 2952.

Wywołanie `/Content/2952/directory.djvu` powoduje pobranie pliku `directory.djvu` z wydania o identyfikatorze 2952.

- `/zipContent` - odpowiedzialny za dostęp do treści publikacji w postaci archiwów ZIP. Wywołanie `/zipContent/2952/` powoduje pobranie treści całego wydania o identyfikatorze 2952 jako pliku zip.
- `/image` - odpowiedzialny za dostęp do miniaturki wydań. Wywołanie `/image?id=2940` powoduje pobranie miniaturki wydania o identyfikatorze 2940.
- `/ka` - odpowiedzialny za szyfrowanie publikacji zabezpieczonych,
- `/rdf` - odpowiedzialny za dostęp metadanych wydań i publikacji w formacie RDF. Wywołanie `/rdf/e2948.xml` powoduje pobranie opisu wydania o identyfikatorze 2948 w formacie RDF. Wywołanie `/rdf/p1651.xml` powoduje pobranie opisu publikacji o identyfikatorze 1651 w formacie RDF.

Część IV. Migracja z wersji 2.2 do wersji 2.5

Ta część opisuje sposób przeprowadzenia migracji systemu dLibra z wersji 2.2 do wersji 2.5 .

Jeżeli zachodzi potrzeba przejścia do wersji 2.5 z wersji starszej niż wersja 2.2 należy najpierw przeprowadzić migrację starszej wersji do wersji 2.2 , a następnie dokonać migracji z wersji 2.2 do 2.5 .

Szczegółową procedurę migracji między wersjami systemu dLibra zawiera Rozdział 11. *Szczegółowa procedura migracji* . Jej trzy główne kroki opisanych w następujących rozdziałach:

Rozdział 12. *Migracja struktur dyskowych komponentów systemu dLibra* Krok ten polega na migracji plików pochodzących z pakietu dystrybucyjnego systemu dLibra do ich nowszej wersji.

Rozdział 13. *Migracja struktur bazy danych* Krok ten polega na wykonaniu zapytań modyfikujących struktury bazy danych.

Rozdział 14. *Migracja danych* Krok ten polega na wykonaniu migracji danych przy pomocy interfejsu administracyjnego serwera systemu dLibra .

Ostrzeżenie

Przy migracji oprogramowania *dLibra* do nowszej wersji zaleca się zachowanie najwyższej ostrożności. Wskazane jest też wykonanie kopii zapasowej całej biblioteki cyfrowej (w tym również treści publikacji) przed rozpoczęciem procesu migracji.

Rozdział 11. Szczegółowa procedura migracji

11.1. Wstęp

W tym rozdziale przedstawiona została zalecana szczegółowa procedura migracji między wersjami systemu *dLibra*. Nie jest ona najkrótszym sposobem przeprowadzenia migracji, ale pozwala na zminimalizowanie okresu niedostępności biblioteki dla użytkowników. Doświadczony administrator systemu *dLibra* może opracować własną procedurę migracji. Niezbędne jednak jest aby zawierała ona kroki opisane w kolejnych rozdziałach.

11.2. Procedura migracji

W celu przeprowadzenia migracji między wersjami systemu *dLibra* zaleca się wykonanie następujących kroków (kolejność jest istotna):

1. Migracja struktur dyskowych komponentów systemu *dLibra* - patrz Rozdział 12. *Migracja struktur dyskowych komponentów systemu dLibra*.
2. Wyłączenie biblioteki cyfrowej - należy wyłączyć Aplikację Czytelnika, a następnie serwer systemu *dLibra*.
3. Wykonanie kopii bezpieczeństwa biblioteki - zaleca się wykonanie kopii bezpieczeństwa katalogu zawierającego treść publikacji i indeksy wyszukiwawcze, katalogu z informacjami generowanymi przez Aplikację Czytelnika (zazwyczaj nazywa się on "*webapp-work*") oraz bazy danych wykorzystywanej przez serwer systemu *dLibra*.
4. Migracja struktur bazy danych - patrz Rozdział 13. *Migracja struktur bazy danych*.
5. Włączenie nowej wersji serwera systemu *dLibra* - należy włączyć *tylko* serwer systemu *dLibra*. Pozwoli to na wykonanie następnego kroku migracji. Nie należy na tym etapie włączać Aplikacji Czytelnika.
6. Migracja danych - patrz Rozdział 14. *Migracja danych*.
7. Włączenie nowej wersji Aplikacji Czytelnika.

Po poprawnym wykonaniu tych kroków system *dLibra* powinien być w pełni zmigrowany do nowej wersji.

Uwaga!

W przypadku wystąpienia problemów na którymkolwiek z etapów migracji prosimy o niezwłoczny kontakt z pomocą techniczną.

Rozdział 12. Migracja struktur dyskowych komponentów systemu *dLibra*

12.1. Przygotowanie wstępnie skonfigurowanych komponentów

Przygotowanie skonfigurowanej postaci uruchamialnej komponentów systemu *dLibra* w nowej wersji powinno zacząć się od użycia instalatora nowej wersji w celu przygotowania tych komponentów w postaci wstępnie skonfigurowanej. Otrzymane w skutek tego komponenty powinniśmy skopiować do katalogu docelowego dla nowej wersji systemu *dLibra*. Zaleca się zastosowanie struktury katalogów analogicznej do przedstawionej poniżej (założono, że poszczególne moduły umieszczone są w podkatalogach pewnego katalogu podstawowego - tutaj `/dlibra/`):

- `/dlibra/dlibra-server- 2.2 /` , `/dlibra/dlibra-webapp- 2.2 /` - katalogi z poprzednią wersją komponentów systemu *dLibra*, która ma zostać zmigrowana do nowej wersji.
- `/dlibra/dlibra-server- 2.5 /` , `/dlibra/dlibra-webapp- 2.5 /` - katalogi z nową wersją komponentów systemu *dLibra*, wstępnie skonfigurowaną przez instalator.
- `/dlibra/dlibra-webapp-work/` - katalog z danymi gromadzonymi przez Aplikację Czytelnika.
- `/dlibra/dlibra-content/` - katalog z treścią i indeksami wyszukiwawczymi.

Następnym etapem tego kroku migracji powinno być przeniesienie zmian samodzielnie wprowadzonych w starej wersji systemu *dLibra* do nowo utworzonej wersji. Zmiany mogą dotyczyć dwóch rodzajów plików - plików binarnych (biblioteki, elementy graficzne itd.) i plików tekstowych (pliki konfiguracyjne, szablony, elementy wyglądu aplikacji czytelnika itd.). Postępowanie w przypadku przenoszenia tych dwóch rodzajów plików opisano w następujących punktach.

12.2. Zmodyfikowane pliki binarne

Nie należy zastępować plików binarnych nowej wersji systemu *dLibra* ich odpowiednikami ze starej wersji. Wyjątkiem są elementy graficzne Aplikacji Czytelnika, które mogły być zmienione w celu dostosowania wyglądu tej aplikacji. W takiej sytuacji należy zmienione pliki przenieść do nowej wersji Aplikacji Czytelnika.

Drugą sytuacją, w której może być konieczne nadpisanie plików binarnych jest wykorzystanie niestandardowych elementów biblioteki **wrapper** w serwerze systemu *dLibra*. Ma to miejsce, gdy system *dLibra* jest uruchamiany w systemach operacyjnych innych niż MS Windows lub Linux. W takiej sytuacji prosimy o kontakt z pomocą techniczną przed rozpoczęciem migracji.

12.3. Zmodyfikowane pliki tekstowe

W przeciwieństwie do plików binarnych, w przypadku zmienionych plików tekstowych nie można ich wprost przenieść ze starej do nowej wersji systemu *dLibra*. Należy wykorzystać narzędzie do porównywania plików (np. *KDiff3* dla Linuxa lub *WinDiff* dla MS Windows) w celu wykrycia różnic pomiędzy plikami i na tej podstawie wprowadzić modyfikacje do nowych plików.

Jeżeli pewnych plików tekstowych brakuje, znaczy to, że informacje w nich zawarte zostały przeniesione w inne miejsce. W przypadku wersji 2.5 jest to część plików konfiguracyjne serwera. Przyczyną zmian jest odseparowanie konfiguracji poszczególnych usług serwera w odrębnych podkatalogach. Zmiany dotyczą następujących plików (wszystkie poniższe ścieżki są względne dla głównego katalogu konfiguracyjnego serwera `conf`):

- plik `dc-pattern.properties` został przeniesiony do podkatalogu `ms`,
- plik `diacritic.properties` został przeniesiony do podkatalogu `se`,
- plik `ldap.properties` został przeniesiony do podkatalogu `us`,
- plik `liusConfig.properties` został przeniesiony do podkatalogu `se`,
- plik `zip_readme.txt` został przeniesiony do podkatalogu `cs`.

Ponadto w podkatalogach konfiguracyjnych poszczególnych usług dodano pliki `jobs.xml` zawierające konfigurację zadań okresowych realizowanych przez te usługi. Informacje znajdujące się w tych plikach w poprzedniej wersji systemu znajdowały się w pliku `server.xml`. Główny plik konfiguracyjny mechanizm zadań okresowych wspólny dla wszystkich usług to plik `quartz.properties` znajdujący się w katalogu `conf`.

Rozdział 13. Migracja struktur bazy danych

W celu wykonania migracji struktur danych należy podłączyć się do bazy danych wykorzystywanej przez serwer dLibra i wykonać zapytania SQL dokonujące migracji. Plik z zapytaniami znajduje się w pakiecie dystrybucyjnym w katalogu `/migrate-sql/[rodzaj.bazy.danych]/`. Plik zapisany jest w kodowaniu UTF-8 i niezbędne jest, aby dane w nim zawarte zostały wprowadzone do bazy danych tak, aby nie nastąpiło przekłamanie związane ze zmianą kodowania znaków. Przykładowe oprogramowanie umożliwiające wprowadzenie takich danych do wielu różnych rodzajów baz danych to bezpłatny `SQuirreL SQL` bazujący na języku Java i protokole JDBC.

Wewnątrz pliku wydzielono sekcje zapytań realizujące kolejne kroki migracji struktur danych. Ze względu na bezpieczeństwo danych zaleca się, aby migracji struktur danych dokonywać stopniowo, zgodnie z sekcjami wewnątrz pliku z zapytaniami.

Jednym z elementów migracji bazy danych w wersji 2.5 jest narzucenie ograniczenia na unikalność adresu e-mail użytkowników. Może się jednak zdarzyć, że nie będzie to możliwe ze względu na zduplikowane adresy e-mail. W takiej sytuacji administrator biblioteki cyfrowej powinien podjąć decyzję co zrobić ze zduplikowanymi adresami. Najprościej w trakcie migracji zmienić je z poziomu bazy danych na tymczasowe unikalne, a po uruchomieniu nowej wersji systemu przy pomocy Aplikacji Administratora wprowadzić docelowe zmiany. Często zdarza się, że zduplikowane adresy e-mail odpowiadają zduplikowanym kontom założonym przez czytelników i wtedy można np. starsze z takich zduplikowanych kont usunąć.

Rozdział 14. Migracja danych

Migracja danych odbywa się poprzez uruchomienie określonych metod z interfejsu administratora serwera systemu *dLibra*. Szczegółowy opis tego interfejsu zawiera Rozdział 7. *Panel administracyjny*.

W ramach migracji między wersją 2.2 a wersją 2.5 należy wykonać następujące zadania:

1. Reindeksacja metadanych - niezbędna ze względu na stworzenie indeksów wyszukiwawczych dla publikacji oraz zmianę sposobu daty w indeksach wyszukiwawczych. Jest ona wykonywana przez uruchomienie polecenia `SearchServer.ReindexTasks.reindexAllEditions(indexMode=5,pubIndexMode=3)`.

Uwaga!

W zależności od wielkości biblioteki cyfrowej i szybkości serwera operacja może trwać nawet kilkanaście godzin.

2. Stworzenie sum kontrolnych dla przechowywanych plików z treścią publikacji - niezbędne dla działania mechanizmu okresowej kontroli spójności treści publikacji. Aby stworzyć sumy kontrolne dla już istniejących plików należy uruchomić polecenie: `ContentServer.DigestTasks.recreateMissingDigests()`. Sumy kontrolne nowych publikacji będą tworzone automatycznie.

Uwaga!

W zależności od wielkości biblioteki cyfrowej i szybkości serwera operacja może trwać nawet kilkanaście godzin.

3. Konwersja haseł użytkowników z postaci czystego tekstu na skrót - jest to operacja opcjonalna zwiększająca bezpieczeństwo systemu. Po jej przeprowadzeniu w bazie danych będą przechowywane tylko skróty haseł użytkowników.

Uwaga!

Po przeprowadzeniu tej operacji odzyskanie pełnych haseł NIE będzie możliwe.

Aby przejść na bezpieczny sposób przechowywania haseł należy:

- uruchomić polecenie `UserServer.AuthorizationTasks.hashUserPasswords()`,
- wyłączyć panel administracyjny serwera i serwer,
- zmienić w pliku konfiguracyjnym serwera `server.xml` w sekcji dotyczącej usługi `UserServer` wartość parametru `plain.authorization.mode` z `true` na `false`,
- włączyć ponownie serwer.

Od tego momentu przy operacji logowania do systemu będą porównywane skróty haseł.

Dodatek A. Parametry konfiguracyjne instalatora systemu *dLibra*

Poniższa lista opisuje informacje niezbędne w trakcie procesu instalacji oprogramowania *dLibra* . Parametry oznaczone jako opcjonalne mogą mieć przypisane puste wartości, przy czym pusta wartość może oznaczać wyłączenie mechanizmu, który wykorzystuje dany parametr.

Ogólne ustawienia serwera

- **serverHostname** - adres komputera, na którym będzie uruchomiony serwer *dLibra* . O ile to możliwe, zaleca się podane zewnętrzny adres serwera (czyli innego niż `localhost` , czy `127.0.0.1`).
- **serverPort** - port, na którym serwer *dLibra* ma oczekiwać na połączenia.
- **contentHome** - katalog, w którym serwer będzie przechowywał zgromadzone publikacje oraz indeksy używane podczas wyszukiwania publikacji..
- **zipArchivePath** (parametr opcjonalny) - URL do archiwum ZIP, w którym mają być przechowywane kopie publikacji umieszczanych w bibliotece cyfrowej. Może być w postaci `file:///...` dla archiwum w ramach lokalnego systemu plików lub `ftp://...` dla archiwum FTP. Ważne jest, aby podana ścieżka kończyła się znakiem `/`.

Konfiguracja bazy danych

- **databaseType** - rodzaj bazy danych, która będzie wykorzystywana przez serwer *dLibra* . Obecnie możliwe wartości tego parametru to: `oracle` , `postgresql` oraz `mysql` .
- **databaseHostname** - adres serwera, na którym uruchomiona jest baza danych wykorzystywana przez serwer *dLibra* .
- **databasePort** - port, na którym serwer bazy danych oczekuje na połączenia (domyślnie 1521 dla Oracle, 5432 dla PostgreSQL oraz 3306 dla MySQL).
- **databaseName** - nazwa bazy danych, z której powinien korzystać serwer systemu *dLibra* .
- **databaseUser** - nazwa użytkownika wykorzystywana podczas autoryzacji w bazie danych.
- **databasePassword** - hasło wykorzystywane podczas autoryzacji w bazie danych.

Konfiguracja serwera SMTP

- **smtpHost** - adres serwera poczty wychodzącej wykorzystywanego do wysyłania wiadomości z aplikacji WWW.
- **smtpAuthorization** - określa, czy serwer pocztowy wymaga autoryzacji podczas wysyłania wiadomości e-mail. Poprawne wartości to `true` lub `false` .

- **smtpUsername** (parametr opcjonalny, wymagany gdy **smtpAuthorization = true**) - nazwa użytkownika używana do autoryzacji na serwerze poczty wychodzącej podczas wysyłania wiadomości z aplikacji WWW.
- **smtpPassword** (parametr opcjonalny, wymagany gdy **smtpAuthorization = true**) - hasło używane do autoryzacji na serwerze poczty wychodzącej podczas wysyłania wiadomości z aplikacji WWW.
- **fromMail** - adres poczty elektronicznej, który ma być wykorzystywany jako adres nadawcy wiadomości przesyłanej przez system *dLibra*.
- **toMail** - adres poczty elektronicznej, na który mają trafiać wiadomości wysyłane z formularza kontakt aplikacji WWW.

Konfiguracja aplikacji WWW

- **webappUrl** - adres URL, pod którym dostępna będzie aplikacja WWW (UWAGA: ze względu na specyfikę technologii Java WebStart nie należy używać adresu `http://localhost/`).
- **webappWorkingDir** - ścieżka do katalogu, w którym przechowywane będą dane wygenerowane podczas działania aplikacji WWW (statystyki dostępu do publikacji, informacje o kontaktach czytelników itp.).
- **webappZipArchivePath** (parametr opcjonalny) - ścieżka do archiwum ZIP z treścią publikacji przechowywanych w bibliotece cyfrowej. Parametr analogiczny do parametru **zipArchivePath**, umożliwiający przyspieszenie dostępu do archiwum. Uwaga: parametr ten powinien wskazywać na to samo archiwum ZIP, które jest ustawione w serwerze *dLibra*.
- **webappCASUrl** (parametr opcjonalny) - adres URL, pod którym dostępny jest system pojedynczego logowania CAS.
- **webappUseCache** - wartość tego parametru określa, czy aplikacja czytelnika ma cache'ować część danych pobieranych z serwera. Użycie mechanizmu cache w przypadku dużych bibliotek cyfrowych przyspieszy działanie stron WWW kosztem aktualności wyświetlanych danych. Poprawne wartości to **true** lub **false**.
- **webappIsServerLocal** - parametr ten pozwala określić, czy serwer i aplikacja czytelnika są uruchomione na tym samym komputerze. Jeżeli tak, to dostęp do treści publikacji realizowany jest bez pośrednictwa serwera, co jest rozwiązaniem wydajniejszym. Poprawne wartości to **true** lub **false**.

Konfiguracja serwera LDAP

Notatka

Obecność pierwszego z poniższych parametrów decyduje o tym, czy pozostałe parametry są wymagane.

- **ldapServerHost** (parametr opcjonalny) - adres serwera LDAP wykorzystywanego do autoryzacji użytkowników.
- **ldapServerPort** (parametr opcjonalny) - port, na którym nasłuchuje serwera LDAP.
- **ldapServerBaseDN** (parametr opcjonalny) - bazowe DN serwera LDAP.

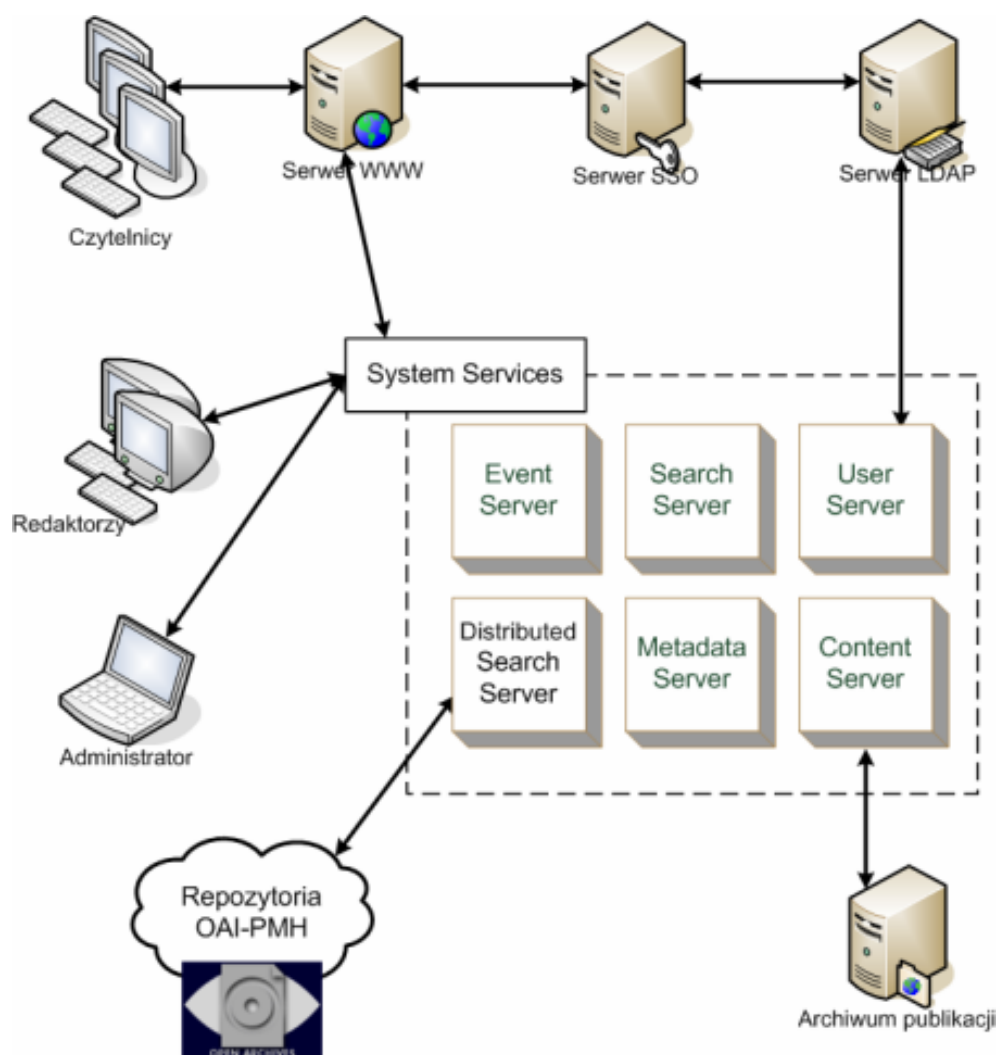
- **ldapAdminDN** (parametr opcjonalny) - DN użytkownika administracyjnego serwera LDAP.
- **ldapAdminPassword** (parametr opcjonalny) - hasło użytkownika administracyjnego serwera LDAP.
- **ldapLoginAttribute** (parametr opcjonalny) - nazwa atrybutu w serwerze LDAP, którego wartość będzie traktowana jako nazwa użytkownika.
- **ldapSearchClassFilter** (parametr opcjonalny) - filtr określający klasy użytkowników w LDAP.
- **ldapAutoFollowReferrals** (parametr opcjonalny) - określa, czy w procesie wyszukiwania powinien być wykorzystywany mechanizm odwołań między serwerami LDAP. Poprawne wartości to **true** lub **false** .

Dodatek B. Architektura oraz możliwości skalowania systemu dLibra

B.1. Architektura systemu dLibra

System dLibra jest systemem wielowarstwowym o architekturze przedstawionej na poniższym schemacie:

Rysunek B.1. Architektura systemu dLibra.



Oznaczony przerywaną linią prostokąt reprezentuje serwer systemu dLibra opisany dokładniej w dalszej części tekstu. Z serwerem komunikują się następujące komponenty architektury systemu:

- *Aplikacja redaktora i administratora* - osobna aplikacja wykorzystywana do zarządzania biblioteką cyfrową, do umieszczania i modyfikacji treści i metadanych dostępnych w bibliotece, etc.
- *Aplikacja czytelnika - uruchomiona na serwerze WWW* aplikacja, dzięki której czytelnicy mogą przeglądać oraz przeszukiwać zawartość biblioteki cyfrowej. Aplikacja WWW może do autentykacji użytkowników wykorzystywać systemy typu Single Sign-On. Aplikacja ta udostępnia również interfejs dostępowy oparty na protokole OAI-PMH.
- *Panel administracyjny serwera dLibra* - panel oparty o technologię Java JMX umożliwiający monitorowanie stanu serwera oraz wykonywanie operacji przewidzianych do obsługi sytuacji awaryjnych. Są to m.in. operacje takie jak odtworzenie indeksów wyszukiwawczych.

B.2. Serwer systemu dLibra

Serwer systemu dLibra składa się z szeregu współdziałających ze sobą usług sieciowych, tworzących razem w pełni funkcjonalną bibliotekę cyfrową. Usługi te to:

- *Content Server* - odpowiada za przechowywanie i udostępnianie treści obiektów umieszczanych w bibliotece cyfrowej. Ma możliwość automatycznego wykonywania kopii bezpieczeństwa na zewnętrzny serwer archiwum publikacji.
- *Metadata Server* - odpowiada za przechowywanie i udostępnianie metadanych na temat obiektów umieszczonych w bibliotece cyfrowej. Zarządza również wewnętrzną strukturą biblioteki, podziałem na katalogi, kolekcje, zestawem atrybutów, słownikami wartości atrybutów itd.
- *Search Server* - odpowiedzialny za indeksowanie i przeszukiwanie treści oraz metadanych na temat obiektów dostępnych w bibliotece cyfrowej.
- *Distributed Search Server* - odpowiedzialny za pobieranie, przechowywanie i udostępnianie metadanych na temat obiektów dostępnych w innych bibliotekach cyfrowych dostępnych poprzez protokół OAI-PMH.
- *User Server* - przechowuje informacje na temat użytkowników, grup użytkowników oraz uprawnień do poszczególnych publikacji. Jest wykorzystywany do autentykacji i autoryzacji przy dostępie do zasobów biblioteki cyfrowej. Może wykorzystywać zewnętrzne bazy użytkowników dostępne poprzez protokół LDAP.
- *Event Server* - wykorzystywany do asynchronicznej komunikacji pomiędzy pozostałymi serwerami systemu dLibra.
- *System Services* - usługa wykorzystywana do łączenia usług systemu dLibra w bibliotekę cyfrową oraz do autoryzacji dostępu pomiędzy usługami.

B.3. Skalowanie systemu dLibra

Każda z usług systemu dLibra może być uruchomiona na osobnym serwerze lub też usługi te mogą być łączone w grupy. Dodatkowo każda z usług systemu dLibra wymaga do swojego działania relacyjnej bazy danych. Usługi mogą współdzielić między sobą jedno konto bazy danych, ale mogą też korzystać z odrębnych baz danych uruchomionych na osobnych serwerach.

Podstawowy wariant konfiguracji systemu dLibra to wszystkie usługi uruchamiane na pojedynczym serwerze jako jedna grupa. W przypadku potrzeby rozbudowy biblioteki cyfrowej istnieje możliwość przenoszenia poszczególnych komponentów systemu na odrębne serwery. Obecnie maksymalny stopień rozbicia systemu dLibra oznacza uruchomienie tego systemu na piętnastu serwerach: 7 x serwer bazy danych dla poszczególnych usług, 7 x serwer dla poszczególnych usług systemu dLibra, 1 x serwer na aplikację czytelnika. Takie rozbicie jest oczywiście uzasadnione tylko w skrajnych przypadkach. Poniżej opisano przykłady przeskalowania systemu dLibra na 2 oraz 3 serwery.

B.3.1. Dwa serwery

Pierwszy serwer:

- Następujące usługi systemu dLibra: Metadata Server, User Server, Event Server, System Services, Content Server
- Serwer WWW z aplikacją czytelnika
- Baza danych współdzielona przez wszystkie usługi pierwszego serwera

Drugi serwer:

- Następujące usługi systemu dLibra: Search Server, Distributed Search Server
- Baza danych współdzielona przez wszystkie usługi drugiego serwera

Takie podejście pozwala wydzielić część systemu dLibra zajmującą się indeksowaniem oraz przeszukiwaniem treści i metadanych. Jest to część, która generuje duże obciążenie serwera w momencie dodawania nowych publikacji do biblioteki cyfrowej. Dzięki temu w momencie maksymalnych obciążeń drugiego serwera, związanych z analizą zawartości obiektów cyfrowych, nadal możliwe jest funkcjonowanie aplikacji czytelnika oraz praca użytkowników aplikacji administratora oraz redaktora, gdyż wszystkie metadane wraz ze stosownymi usługami znajdują się na pierwszym serwerze.

B.3.2. Trzy serwery

Pierwszy serwer:

- Następujące usługi systemu dLibra: Metadata Server, User Server, Event Server, System Services, Content Server
- Baza danych współdzielona przez wszystkie usługi pierwszego serwera

Drugi serwer:

- Następujące usługi systemu dLibra: Search Server, Distributed Search Server
- Baza danych współdzielona przez wszystkie usługi drugiego serwera

Trzeci serwer:

- Serwer WWW z aplikacją czytelnika

Takie podejście dodatkowo odseparowuje aplikację czytelnika, która może powodować okresowe większe obciążenia w przypadku zwiększonego ruchu czytelników. Dzięki

wbudowanemu w aplikację czytelnika mechanizmowi cache, jest ona stosunkowo niezależna od usług systemu dLibra, a więc takie przeskalowanie systemu umożliwia rozłożenie na dwa serwery (odpowiednio pierwszy i trzeci) obciążenia związanego z pracą użytkowników aplikacji redaktora i administratora oraz obciążenia związanego z obsługą żądań czytelników. Podobnie jak w poprzednim scenariuszu, drugi serwer odpowiedzialny jest za indeksowanie i przeszukiwanie udostępnianych treści cyfrowych i metadanych.