



MILESS 2.1
Administrations-Handbuch
Installation und Konfiguration

Stand: 06.10.2006

Frank Lützenkirchen
luetzenkirchen@ub.uni-duisburg-essen.de

Inhaltsverzeichnis

Inhaltsverzeichnis.....	3
Einführung: MILESS, DuEPublico und MyCoRe.....	5
Das MILESS Datenmodell.....	7
Systemanforderungen und benötigte Software.....	8
Herunterladen der Software	10
Einrichten von Java und Ant	11
Übersetzen des Java-Quellcodes	13
Einrichten der Datenbank.....	13
Konfiguration der Datenbankverbindung	14
Anlegen der Tabellen.....	14
Herunterladen der Beispielinhalte.....	15
Verwendung des Command Line Interface.....	16
Laden der Beispiel-Klassifikationen.....	16
Laden der Personendaten	18
Verzeichnisse für Suchindex konfigurieren.....	19
Laden der Dokumenten-Metadaten.....	19
Erkennung von Dateitypen konfigurieren.....	20
Speicherorte für Dateien konfigurieren.....	20
Integration eines Helix Universal Streaming Servers.....	21
Integration eines IBM VideoCharger Servers.....	23
Auswahlregeln für Speicherorte anpassen	24
Beispiel-Dateien laden	25
Web-Applikation erstellen	25
Apache Tomcat konfigurieren	25
Integration von Tomcat in den Apache Webserver	27
Email-Benachrichtigung konfigurieren.....	28
Zugriffsstatistiken für einzelne Dokumente.....	29
Personenrollen konfigurieren	30
Beziehungstypen zwischen Dokumenten konfigurieren	31
Wartungsmodus	31
Logging konfigurieren.....	32
Datei-Upload konfigurieren	33
Download von Videos zulassen	35
Verteilte Suche über mehrere Server.....	36
Datensicherung und Suchindex.....	37
Zugang für Internet-Suchmaschinen	38
Bilder und CSS anpassen	39
XSL Stylesheets anpassen.....	39
Anpassen der Navigationsstruktur.....	42
Webseiten, Suchmasken und Eingabeformulare anpassen.....	43
Benutzerverwaltung	43
Rechteverwaltung	45
Publikationsworkflow anpassen.....	47
Persistent Identifier und URN-Vergabe.....	51
Hochschulschriften an die DNB melden	53
OAI Data Provider konfigurieren	54

Web Services Schnittstellen	56
SCORM-Lernmaterial suchen und anzeigen.....	57
Semesterapparate online	58

Einführung: MILESS, DuEPublico und MyCoRe

MILESS ist ein auf Java- und XML-Technologien basierender institutioneller Dokumenten- und Publikationsserver. Neben Texten wie Hochschulschriften, Vorlesungsskripten und Dissertationen können auch Animationen und Audio- oder Videodateien (Real, WMV, AVI, MP3, MPEG, Quicktime) über einen integrierten Streaming Server verteilt werden. Autoren stellen direkt über ihren Webbrowser Inhalte in MILESS ein, die dann über konfigurierbare Suchmasken, über Browsing in hierarchischen Klassifikationssystemen (beispielsweise nach Organisationsstruktur oder Dewey-Klassifikation) oder über einen Personenindex von A-Z auffindbar sind. Über das Modul "Online-Semesterapparate" können Dozenten Literaturlisten, Links, digitalisierte Volltexte von Zeitschriftenartikeln und Buchauszügen sowie eigene Dateien bereitstellen. Dozenten pflegen die Inhalte über einfache Formulare, die in der Bibliothek die notwendigen Dienstleistungen wie Buchausleihe oder Digitalisierung anstoßen. Kenntnisse zur Gestaltung von Internetseiten sind nicht erforderlich.

Alle MILESS Dokumente sind mit einem Titelblatt versehen, das Metadaten im Dublin-Core-Standard enthält. Über mehrere Schnittstellen können die in MILESS bereitgestellten Dokumente auch weltweit verfügbar gemacht. MILESS verfügt über eine Schnittstelle für das Metadata-Harvesting-Protokoll der Open-Archives-Initiative (OAI-PMH 2.0). Die Metadaten und Volltexte von Dissertationen und anderen Publikationen können über diese Schnittstelle (OAI Data Provider) bereitgestellt, national und international durch OAI Service Provider gesammelt und damit auch auf internationaler Ebene recherchierbar gemacht werden. Weiterhin ist ein direkter Zugang für Suchmaschinen-Robots realisiert, über den MILESS-Inhalte durch Internet-Suchmaschinen wie Google gefunden und indiziert werden.

Neue Funktionen in der Version 2.1:

- Web Services zur Integration in externe Systeme, z. B. Lern-Management-Systeme: Erstellen, Bearbeiten, Upload und Suche in Inhalten über SOAP-Schnittstellen
- Archivierung von SCORM-Modulen: Inhaltsbasierte Suche in den Metadaten von E-Learning-Paketen, integrierter SCORM Viewer
- Vergabe von Persistent Identifiern (URNs, urn:nbn:de) für Publikationen und Meldung an die Deutsche Nationalbibliothek über EMail oder OAI-Schnittstelle
- Workflow zur Freigabe von Publikationen
- Animierte Tutorials für Autoren
- Individuelle Zugriffsstatistik für jedes einzelne Dokument
- Online-WYSIWYG-HTML-Editor für Semesterapparate

MILESS ist Open Source unter der GNU General Public License. MILESS kann allein auf Basis einer Open Source Umgebung ohne jegliche Kosten für Software-Lizenzen betrieben werden und ist auf Windows, Linux,

Solaris und IBM AIX lauffähig. Als Datenbank kann MySQL, Oracle oder IBM DB2 verwendet werden. Weiterhin ist nur ein Web Application Server, z. B. Jakarta Tomcat (Open Source) erforderlich.

Dieses Dokument beschreibt die Installation und Konfiguration von MILESS. Im Anschluss daran müssen Sie dann insbesondere das Layout der Oberfläche ggf. an Ihre Einrichtung anpassen. Nach der MILESS Installation sehen Sie den Dokumentenserver so, wie er an der Universität Duisburg-Essen betrieben wird. Dort ist MILESS (ursprünglich eine Entwicklung der inzwischen mit Duisburg fusionierten Universität Essen) im August 2006 mit dem Duisburger Dokumentenserver DuetT fusioniert; die DuetT Inhalte wurden in das MILESS System übernommen. Der neue, gemeinsame Server auf Basis der MILESS Software heißt nun DuEPublico¹ – Duisburg-Essen Publications Online. MILESS ist nun also der Name der Software, DuEPublico der Name der lokalen Instanz in Duisburg-Essen.

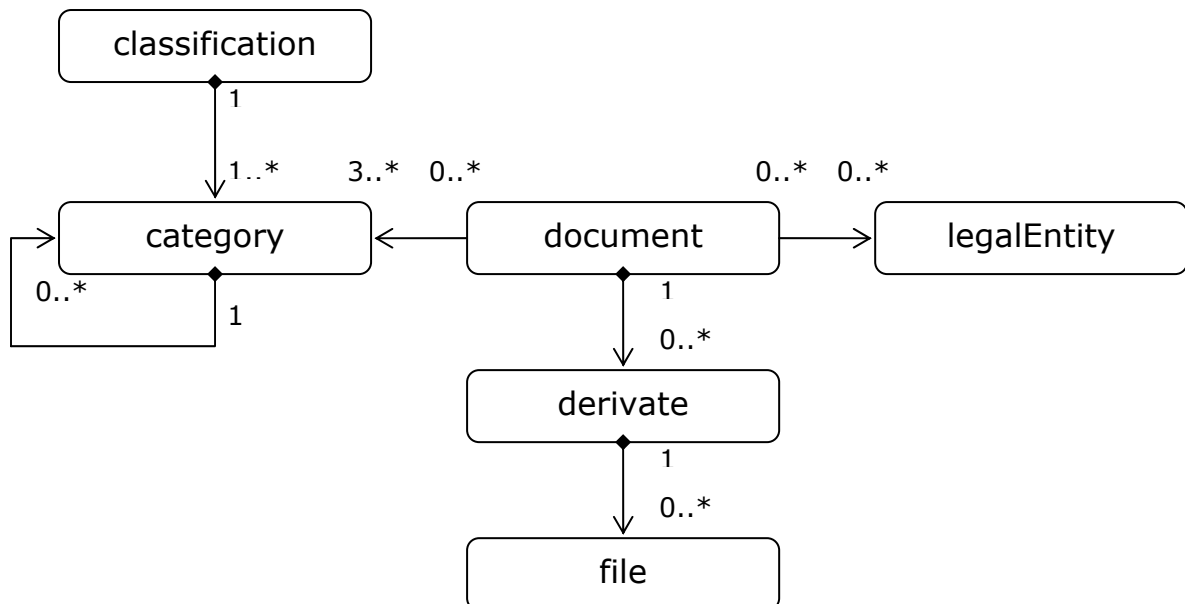
MILESS wird zur Zeit ebenfalls an den Universitäten Münster, Freiburg, Bochum, Jena, Paderborn und Düsseldorf eingesetzt. Aus der Gruppe dieser Nachnutzer hat sich ein detaillierteres Bild von generellen Anforderungen an Dokumenten- und Publikationsserver, Gemeinsamkeiten und Unterschieden ergeben. Aus dieser Zusammenarbeit und den Bedürfnissen der einzelnen beteiligten Einrichtungen heraus ist die MyCoRe-Initiative² entstanden, die einen gemeinsamen Anwendungskern für jeweils lokale Applikationen entwickelt, den MyCoRe Kern. MyCoRe entwickelt für Dokumenten-, Publikations- und Archivserver (oder allgemeiner "Content Repositories", daher das "CoRe") einen gemeinsamen Software-Kern ("core"), der sich an die eigenen Bedürfnisse anpassen und erweitern lässt (daher das "My" für die lokale Adaption). Auf der Basis dieses Kerns, der wie MILESS als Open Source unter der GNU General Public License verfügbar ist, können spezifische Anwendungen an den jeweiligen Standorten schneller realisiert werden. Auch MILESS baut inzwischen zu ca. 50 % auf dem MyCoRe Code auf, beteiligt sich an der gemeinsamen Weiterentwicklung und profitiert wesentlich von vielen MyCoRe-Funktionen, die an anderen Universitäten entwickelt wurden.

¹ <http://duepublico.uni-duisburg-essen.de/>

² <http://www.mycore.de/>

Das MILESS Datenmodell

Die folgende Abbildung gibt einen Überblick über das MILESS Datenmodell.



Ein **Document** besitzt inhaltliche, den Elementen des Dublin Core Datenmodells entsprechende Metadaten wie Titel, Abstract oder Stichwörter. Es ist das zentrale Element des MILESS Datenmodells. Autoren sind in MILESS nicht als Namen in den Dokumenten eingetragen, sondern als Verweise auf eigenständige Datensätze, auf LegalEntities.

Ein **LegalEntity** ist eine Person oder Körperschaft mit eigenen Daten wie dem Namen, insbesondere aber Kontaktdaten wie Telefon, Email, Anschrift usw. Dokumente verweisen auf LegalEntities, die die Rolle von Autoren, Herausgebern oder sonstigen Beitragenden zu Dokumenten spielen.

Jedes Dokument besitzt ein oder mehrere Dateibereiche, die Derivate. Ein **Derivat** ist eine Menge von logisch zusammengehörenden Dateien, z. B. können die HTML-Version, die PDF-Version und die Dateien des Ursprungsformates LaTeX einer Dissertation jeweils ein eigenes Derivat bilden. MILESS erlaubt es so, die Dateien verschiedener Formate gemeinsam unter einem Dokument einzuordnen. Derivate können auch nur aus einem Link auf eine externe Webseite (externes Derivat) bestehen.

Eine **Klassifikation** ist eine Listen- oder Baumstruktur von Kategorien. Diesen Kategorien können MILESS Dokumente zugeordnet werden. Beispiele für solche Klassifikationen sind die MILESS Dokumententypen, Medientypen und die Hierarchie der Fachbereiche und Arbeitsgruppen (Organisationsstruktur), aber auch fachliche Klassifikationen wie Dewey

Decimal Classification (DDC) oder PACS (Physics and Astronomy Classification Scheme). Jedes Dokument ist mindestens in die Klassifikationen TYPE, FORMAT und ORIGIN eingeordnet.

Titel:	Die semiklassische Rolle periodischer Bahnen im gemischten Phasenraum	
URL für Bookmark:	http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=141	
URN (NBN):	urn:nbn:de:hbz:465-miless-000141-1	
Kollektion:	Dissertationen	
Status:	Dokument veröffentlicht	
Sprache:	Deutsch	
Dokumententyp:	Wissenschaftliche Abschlussarbeiten » Dissertation	
Medientyp:	Text	
Autor:	Schomerus, Henning [Autor]	
Dateien:	[Dateien anzeigen] [Details] [ZIP-Datei erzeugen]	Adobe PDF 5,45 MB in einer Datei [PlugIn/Viewer Download]
	[Dateien anzeigen] [Details] [ZIP-Datei erzeugen]	HTML-Dokument, TeX DVI-Druckdatei, PostScript 11,08 MB in 1272 Dateien [PlugIn/Viewer Download]
	[Dateien anzeigen] [Details] [ZIP-Datei erzeugen]	LaTeX Dokument (Quellcode), PostScript 6,16 MB in 68 Dateien [PlugIn/Viewer Download]
	[Dateien anzeigen] [Details] [ZIP-Datei erzeugen]	PostScript 6,74 MB in einer Datei [PlugIn/Viewer Download]
Stichwörter:	Semiklassische Methoden, Gemischter Phasenraum, Periodische Bahnen, Bifurkationstheorie, Chaostheorie	
Klassifikation Physik:	Quantentheorie Nichtlineare Dynamik	
Physics and Astronomy Classification Scheme:	00. General » 03. Classical and quantum physics: mechanics and fields » 03.65.-w Quantum mechanics » 03.65.Sq Semiclassical theories and applications	
Dewey Dezimal-Klassifikation:	500 Naturwissenschaften und Mathematik » 530 Physik	

Abbildung 1: Beispiel einer Dokumenten-Titelseite

Systemanforderungen und benötigte Software

- **Betriebssystem:**
MILESS ist nachweislich unter Sun Solaris 9 oder höher, Windows XP, Linux und IBM AIX 5 oder höher lauffähig.
- **Datenbank:**
Sie benötigen ein relationales Datenbanksystem, das eine JDBC-Schnittstelle (Java Database Connectivity) unterstützt. MILESS wurde getestet mit MySQL 4.0 oder höher, IBM DB2 8.2 oder höher sowie Oracle 8 oder höher. Grundsätzlich sollte die Software nach geringen Anpassungen auch mit jeder anderen Datenbank lauffähig sein.
- **Java:**
Sie benötigen ein Java 2 SDK, Standard Edition, Version 1.4.2 oder

höher, d. h. es muss ein Java-Compiler auf Ihrem System installiert sein. Diesen können Sie von den Webseiten der Firma Sun³ kostenfrei herunterladen.

- **Apache Ant:**

Für die Installation wird das Java-Build-Werkzeug Apache Ant Version 1.6.2 oder höher benötigt, das Sie ebenfalls kostenfrei herunterladen können.⁴

- **Servlet Container:**

MILESS ist eine Java-Servlet-Anwendung. Sie benötigen daher einen Servlet Container wie Apache Tomcat 5.0⁵, können aber auch andere Servlet Container wie etwa IBM WebSphere oder Jetty⁶ verwenden. Apache Tomcat ist robust, stabil und kann ebenfalls kostenlos als Open Source heruntergeladen werden.

Die folgende Software ist optional und kann zusätzlich eingesetzt werden:

- **Webserver:**

Dieser kann dem Tomcat Servlet-Container als "Reverse Proxy" vorgelagert werden, was im Produktionsbetrieb verschiedene Vorteile hat (z.B. sichererer Betrieb ohne "root"-Rechte). Wir empfehlen den Apache Webserver 2.0 oder höher.

- **IBM Content Manager:**

Alternativ zu herkömmlichen Dateisystemen auf dem Server können in MILESS archivierte Dateien auch in dem kommerziellen Dokumentenmanagementsystem IBM Content Manager (Version 8 oder höher) gespeichert werden.

- **Video Streaming Server:**

Wenn Sie einen Real Server 8, Helix Universal Server 9 oder höher, oder einen IBM VideoCharger Server Version 7 oder höher haben, kann MILESS Audio-/Video-Dateien automatisch (d. h. für den Anwender transparent) auf diesen Servern speichern und von dort streamen.

- **Java Service Wrapper⁷:**

Diese Software wird für den Produktionsbetrieb empfohlen. Sie ermöglicht den Betrieb der Java-Anwendung als Dienst, vereinfacht die Überwachung und startet z.B. die Anwendung in Fehlerfällen automatisch neu.

- **CVS Client:**

Über einen CVS Klienten (Concurrent Versions System⁸) können Sie MILESS direkt im Quellcode aus dem Versionsmanagementsystem herunterladen, so dass eine Aktualisierung der Software einfacher ist.

³ <http://java.sun.com/>

⁴ <http://ant.apache.org/>

⁵ <http://tomcat.apache.org/>

⁶ <http://www.mortbay.org/>

⁷ <http://wrapper.tanukisoftware.org/>

⁸ <http://www.cvs-home.org/>

Herunterladen der Software

Wenn Sie einen CVS-Klienten installiert haben, können Sie mit dem folgenden Befehl MILESS direkt aus dem Versionsmanagementsystem herunterladen:

```
cvs -d :pserver:anoncvs@server.mycore.de:/cvs checkout miless
```

Alternativ können Sie MILESS als ZIP-Paket über CampusSource⁹ oder den DuEPublico¹⁰ Server herunterladen. Entpacken Sie den Inhalt des Archivs `miless-2.1.zip` (Windows) bzw. `miless-2.1.tar.gz` (Unix-Systeme) auf Ihrem Rechner in ein neues, leeres Verzeichnis.

Es entsteht jeweils ein Unterverzeichnis `miless/`, das im Folgenden als Installationsverzeichnis bezeichnet wird. Wechseln Sie nun in dieses Verzeichnis. Die Verzeichnisstruktur sieht wie folgt aus:

<code>build.xml</code>	Apache Ant Build-Datei
<code>config</code>	Konfigurationsdateien
<code>documentation</code>	Dokumentation
<code>lib</code>	Verwendete Java-Bibliotheken (*.jar)
<code>module-dozbib</code>	Universitätbibliographie Online (Prototyp, bitte ignorieren)
<code>module-dso</code>	DSO Viewer, ein Bildbetrachter für Digitalisate
<code>module-duett</code>	Migrationsprogramme für DuetT (bitte ignorieren)
<code>module-oai</code>	Open Archivs Initiative Schnittstelle (OAI-PMH 2.0)
<code>module-scorm</code>	SCORM-Unterstützung: Viewer, Konfiguration, Suchmasken
<code>module-semapp</code>	Modul Semesterapparate Online
<code>module-video</code>	Spezielle Audio-/Video-Funktionen wie z. B. Storyboards
<code>module-webservices</code>	WebServices auf Basis von Apache Axis (SOAP)
<code>setup</code>	Installationsskripte, insbesondere für die Datenbanktabellen
<code>sources</code>	Java-Quellcode
<code>stylesheets</code>	XSL Stylesheets, erzeugen aus XML-Ausgaben HTML
<code>web</code>	Webseiten, Grafikdateien und CSS

Einige Teile von MILESS, die Erweiterungen der eigentlichen Kernanwendung darstellen, sind zu Modulen zusammengefasst und in Unterverzeichnissen `module-*` abgelegt. Die Unterverzeichnisse `module-duett` und `module-dozbib` können Sie ignorieren. Jedes Modul ist einheitlich in die folgenden Unterverzeichnisse unterteilt:

<code>build.xml</code>	Apache Ant Build-Datei
<code>conf</code>	Konfigurationsdateien

⁹ <http://www.campussource.de/>

¹⁰ <http://duepublico.uni-duisburg-essen.de/about/nachnutzung.xml>

src Java-Quellcode
web Webseiten, Grafikdateien und CSS
xsl XSL Stylesheets, erzeugen aus XML-Angaben HTML

Einrichten von Java und Ant

Prüfen Sie zunächst, ob die Umgebungsvariable `JAVA_HOME` auf Ihrem System gesetzt ist:

```
echo $JAVA_HOME                   (Unix-Systeme)
```

bzw.

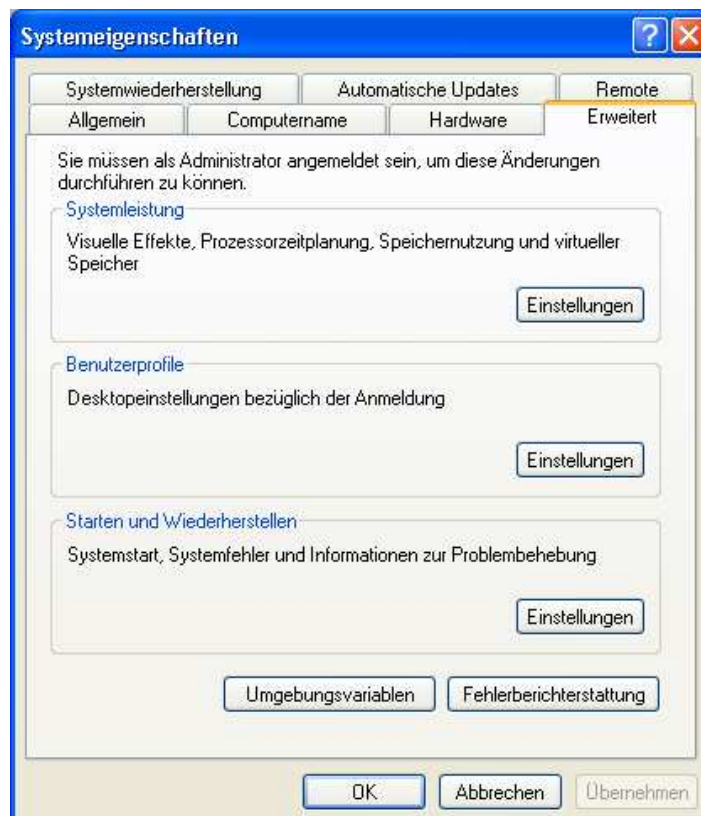
```
echo %JAVA_HOME%                   (Windows)
```

Falls nicht, setzen Sie diese Umgebungsvariable auf das Verzeichnis, in dem Ihre Java-Entwicklungsumgebung (SDK) installiert ist. Auf Unix-Systeme geschieht dies z. B. über den Befehl

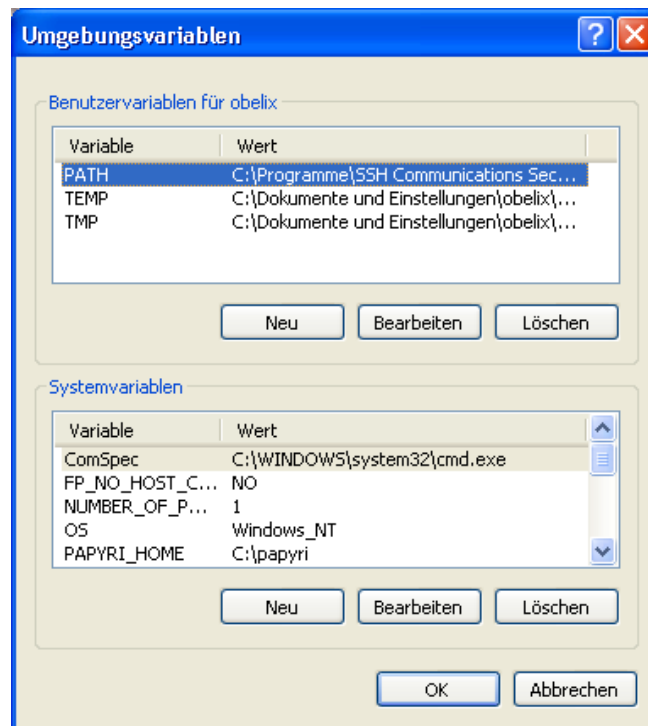
```
export JAVA_HOME=/usr/lib/java
```

Diese Zeile sollten Sie in die Startdatei Ihrer Shell (z.B. `.bashrc` bei Verwendung der BASH Shell) eintragen.

Um eine Umgebungsvariable unter Windows zu setzen, gehen Sie wie folgt vor: Klicken Sie in der Systemsteuerung unter "System" auf "Erweitert":



Nun klicken Sie auf "Umgebungsvariablen", dann auf "Systemvariablen", "Neu", um eine Umgebungsvariable hinzuzufügen:



Das Unterverzeichnis `bin/` Ihrer Java-Installation sollte im Ausführungspfad (Umgebungsvariable `PATH`) eingetragen sein. Sie können durch Aufruf von

```
java -version
```

die installierte Java-Version ermitteln. Weiterhin muss die Umgebungsvariable `ANT_HOME` auf das Verzeichnis gesetzt sein, in dem Sie Apache Ant installiert haben. Auch hier muss das Unterverzeichnis `bin/` der Ant-Installation in der Variable `PATH` eingetragen sein.

Wenn die Installation von Java und Ant korrekt ist, können Sie im MILESS-Verzeichnis "`ant info`" aufrufen und sollten eine Ausgabe wie folgt erhalten:

```
C:\Java\workspace\miless>ant info
Buildfile: build.xml
```

```
paths:
info:
```

```
[echo] Ant version: Apache Ant version 1.6.2 compiled on July 16 2004
[echo] Java JDK version: 1.4.2_07
[echo] Operating system: Windows XP, version: 5.1, architecture: x86
[echo] Xerces XML parser:
[java] Xerces-J 2.6.2
```

```
BUILD SUCCESSFUL
Total time: 4 seconds
```

In den folgenden Schritten wird Apache Ant verwendet, um MILESS aus dem Quellcode zu übersetzen und zu installieren. Dies geschieht durch Aufruf von "targets", wie z. B. "ant info". Wenn Sie im Installationsverzeichnis einfach nur "ant" aufrufen, erhalten Sie eine Liste der verfügbaren Build-Targets.

Übersetzen des Java-Quellcodes

Sie können nun durch Aufruf von

```
ant compile
```

den MILESS-Quellcode mit dem Java-Compiler übersetzen lassen. Es entsteht nun ein neues Verzeichnis `build` mit einem Unterverzeichnis `classes`, das die übersetzten Java-Klassen enthält. Das `build`-Verzeichnis enthält grundsätzlich alle generierten Dateien, die spätere fertige Webanwendung, die zur Laufzeit aktiven Konfigurationsdateien usw. Sie können durch Aufruf von

```
ant clean
```

das komplette `build`-Verzeichnis entfernen und so ggf. die Installation neu beginnen. Der Aufruf

```
ant javadocs
```

erzeugt die HTML-Seiten der JavaDoc-Dokumentation zum MILESS Quellcode im Verzeichnis `build/javadocs/`.

Einrichten der Datenbank

Für den Betrieb von MILESS ist eine relationale Datenbank erforderlich. Bei Verwendung von MySQL sollten Sie eine neue Datenbank "miless" anlegen, zusammen mit einer Benutzerkennung "miless", die Zugriff auf diese Datenbank hat. Details entnehmen Sie bitte der Dokumentation zu MySQL. Die beiden wesentlichen Schritte sind:

```
mysql:> create database miless
mysql:> grant all privileges on miless.* to miless@localhost
      identified by 'password' with grant option
```

Für IBM DB2 können diese beiden Schritte über die folgenden Kommandos ausgeführt werden (Benutzerkennung "miless" muss auf dem Server bereits existieren):

```
db2 => create database miless using codeset UTF-8 territory DE
      collate using system
```

```
db2 => grant dbadm on database to user miless
```

Für Oracle entnehmen Sie bitte der Dokumentation des Produktes, wie dort eine neue Datenbank angelegt wird.

Konfiguration der Datenbankverbindung

Die zentrale Konfigurationsdatei von MILESS ist die Datei `conf/mycore.properties`. In dieser Datei müssen Sie zunächst die JDBC-Verbindung zu der zuvor angelegten Datenbank konfigurieren. Sie finden in `mycore.properties` Vorlagen für die Konfigurationsparameter von MySQL, DB2 und Oracle. Passen Sie diese Parameter an bzw. kommentieren Sie die Parameter für die jeweils nicht benutzte Datenbank aus. Details zur JDBC-Konfiguration können Sie auch der Dokumentation des jeweiligen Datenbanksystems entnehmen. Die folgenden Eigenschaften müssen angepasst werden:

<code>MCR.persistence_sql_database_type</code>	Mögliche Werte: <code>db2 oracle mysql</code>
<code>MCR.persistence_sql_driver</code>	zu verwendender JDBC-Treiber
<code>MCR.persistence_sql_database_url</code>	JDBC Datenbank-URL (<code>jdbc:...</code>)
<code>MCR.persistence_sql_database_userid</code>	Benutzerkennung, falls nicht Teil der URL
<code>MCR.persistence_sql_database_passwd</code>	Passwort, falls nicht Teil der URL

Im darauf folgenden Abschnitt der `mycore.properties` Datei können Sie ggf. die Tabellennamen anpassen, die in der Datenbank verwendet werden. Normalerweise müssen Sie diese jedoch nicht ändern. Wenn Sie die Tabellennamen ändern, haben Sie aber z. B. die Möglichkeit, mehrere MILESS Instanzen in der gleichen Datenbank parallel zu betreiben und die Datenhaltung untereinander zu separieren. Wenn Sie Änderungen an den Tabellennamen vornehmen, müssen Sie die geänderten Tabellennamen auch an folgenden Stellen anpassen:

- In allen Konfigurationseigenschaften mit Namen `MCR.IndexBrowser.*.Table`
- In den SQL-Skripten zur Tabellenerzeugung (`*.sql`) im Unterverzeichnis `setup`

Anlegen der Tabellen

Nun können Sie die Datenbanktabellen anlegen. Rufen Sie

```
ant tables
```

auf. Es werden nun automatisch alle erforderlichen Datenbanktabellen und Indizes erzeugt. Eine Übersicht über die Struktur der Tabellen finden Sie im Verzeichnis `setup/*.sql`. Hier sind für MySQL, DB2 und Oracle jeweils

SQL-Skripte zum Erzeugen, Löschen und Exportieren der Tabellen vorhanden. Die folgenden Tabellen werden angelegt:

DOCUMENTS	Metadaten von Dokumenten (Titel, Autor usw.)
DOCREL	Beziehungen zwischen Dokumenten (z.B. "ist neue Version von ...")
DERIVATES	Dateibereiche (Derivate = logisch zusammengehörige Menge von Dateien)
FILES	Metadaten von Dateien (Pfad, Dateityp, Größe, MD5 Prüfsumme etc.)
MCRFSNODES	Metadaten von Dateien des MyCoRe Internal Filesystem (IFS)
LENAMES	Namen von Personen und Körperschaften
LEXML	Metadaten von Personen und Körperschaften (Anschrift, EMail etc.)
CLASSIF	Klassifikationen, z.B. Fachbereich, Dokumententyp, DDC-Klassifikation
CATEG	Kategorien innerhalb der Klassifikationen: Baum- oder Listenstruktur
CATMEMB	Welche Dokumente sind welchen Kategorien zugeordnet?
MCRURN	Persistent Identifier (NBN URNs: urn:nbn:de:...)
USERORGROUP	Login-Benutzer und Gruppen
USERGROUPIDS	Welche Benutzer sind Mitglied welcher Gruppen?
RIGHTS	Zugriffsrechte von Benutzern oder Subnetzen auf Dokumente

Die Tabellen sind zunächst leer. In den Tabellen `USERORGROUP` und `USERGROUPIDS` werden bereits automatisch einige Standard-Benutzer und erforderliche Gruppen eingetragen.

Herunterladen der Beispielinhalte

Laden Sie nun die Beispielinhalte als ZIP-Paket `content-2.1.zip` bzw. `content-2.1.tar.gz` über CampusSource¹¹ oder den DuEPublico¹² Server herunter und entpacken Sie den Inhalt in ein leeres lokales Verzeichnis. Dieses Paket enthält Beispiele für Klassifikationen, Dokumente, Personeneinträge und Dateien. Die Metadaten sind dabei als XML-Dateien gespeichert, die Sie in MILESS importieren können. Nach dem Entpacken entstehen die folgenden Verzeichnisse:

<code>documents</code>	Metadaten von Dokumenten, in XML-Dateien mit Namen <code>document-*.xml</code>
<code>legalentities</code>	Metadaten von Personen, in XML-Dateien mit Namen <code>legalentity-*.xml</code>

¹¹ <http://www.campussource.de/>

¹² <http://duepublico.uni-duisburg-essen.de/about/nachnutzung.xml>

`classifications` Klassifikationen wie Dokumententypen, Fachbereiche, DDC etc.
in XML-Dateien mit Namen `*.xml`

`derivates` Derivate (Dateikollektionen) in XML-Dateien `derivate-*.xml`
mit Unterverzeichnissen für die eigentlichen Dateien (PDFs etc.)

Verwendung des Command Line Interface

Für administrative Zwecke wie dem Importieren und Exportieren von Inhalten besitzt MILESS eine Kommandozeilen-Schnittstelle. Dazu müssen Sie zunächst einmalig

```
ant bin
```

aufzurufen. Im Verzeichnis `build/bin` ist nun ein Skript `miless.cmd` (Windows) bzw. `miless.sh` (Unix) entstanden. Wenn Sie dieses Skript aufrufen, startet das Command Line Interface mit einem Eingabe-Prompt:

```
miless:>
```

Durch Eingabe von `quit` oder `exit` verlassen Sie das Command Line Interface. Wenn Sie `help` eingeben, erhalten Sie eine Liste der möglichen Kommandos.

Laden der Beispiel-Klassifikationen

Sie können nun die ersten Inhalte in Ihre MILESS-Installation laden. Klassifikationen sind Listen oder Baumstrukturen von Kategorien. In diese Kategorien können Sie MILESS Dokumente einordnen. Benutzer können dann durch die Klassifikationen browsen, um enthaltene Dokumente anzuzeigen, oder die Klassifikationen bei der Suche nutzen. Beispiele für solche Klassifikationen sind die Listen der MILESS Dokumententypen, Medientypen und die Hierarchie der Fachbereiche und Arbeitsgruppen (Organisationsstruktur), aber auch fachliche Klassifikationen wie Dewey Decimal Classification (DDC) oder PACS (Physics and Astronomy Classification Scheme). MILESS erlaubt es, beliebige solche Klassifikationssysteme über eine XML-Datei in das System zu laden und für die Einordnung von Dokumenten zu verwenden. Geben Sie dazu ein

```
miless:> load all classifications from directory classifications
```

oder alternativ

```
miless.cmd|miless.sh load all classifications from directory  
classifications
```

wobei Sie *classifications* durch den Pfad des Verzeichnisses ersetzen, in dem die entpackten Beispiel-Klassifikationen liegen. Es werden die folgenden Klassifikationen aus XML-Dateien gelesen und im System in den SQL-Tabellen `CLASSIF` und `CATEG` gespeichert:

TYPE	Liste der Dokumententypen, z.B. Dissertation, Vorlesungsskript usw.
FORMAT	Liste der Medientypen, z.B. Text, Audio, Video usw.
ORIGIN	Fachbereiche und Einrichtungen: Die Organisationsstruktur der Universität
DDC	Dewey Decimal Classification, drei Ebenen mit insgesamt 1000 Kategorien
PACS	Physics and Astronomy Classification Scheme, Version 1997

Insbesondere die Klassifikation `ORIGIN` werden Sie an Ihre Einrichtung anpassen müssen. Ändern Sie dazu die Datei `ORIGIN.xml` ab. So lange noch keine Dokumente in eine Klassifikation eingeordnet sind, oder Sie nur die Bezeichnungen von Kategorien ändern oder neue Kategorien hinzufügen, reicht ein Aufruf von

```
miless:> update classification from file ORIGIN.xml
```

um die Klassifikation im System zu aktualisieren. Falls später bereits Dokumente in Kategorien eingeordnet sind, können Sie über das Command Line Interface auch Dokumente zwischen Kategorien verschieben. Dazu dienen die Kommandos

```
move documents in classification {0} from category {1} to category {2}
move documents in classification {0} category {1} to classification {2}
category {3}
```

wobei für `{0}`, `{1}` usw. die IDs der Klassifikationen und Kategorien einzusetzen sind.

Die Datei `documentation/xml/classification.xsd` enthält das W3C XML Schema für Klassifikationen.

Dewey Dezimal-Klassifikation (Insgesamt 534 Dokumente nach dieser Klassifikation eingeordnet)

- + [__21 Dok.] 000 Informatik, Informationswissenschaft, allgemeine Werke
- + [__6 Dok.] 100 Philosophie und Psychologie
- + [__1 Dok.] 200 Religion
- + [__44 Dok.] 300 Sozialwissenschaften
- + [__5 Dok.] 400 Sprache
- [301 Dok.] 500 Naturwissenschaften und Mathematik
 - + [__0 Dok.] 500 Naturwissenschaften
 - + [__41 Dok.] 510 Mathematik
 - + [__0 Dok.] 520 Astronomie
 - + [__39 Dok.] 530 Physik
 - + [120 Dok.] 540 Chemie
 - + [__8 Dok.] 550 Geowissenschaften
 - + [__0 Dok.] 560 Paläontologie
 - + [__92 Dok.] 570 Biowissenschaften; Biologie
 - + [__1 Dok.] 580 Pflanzen (Botanik)
 - + [__2 Dok.] 590 Tiere (Zoologie)
- + [188 Dok.] 600 Technik, Medizin, angewandte Wissenschaften
- + [__12 Dok.] 700 Künste und Unterhaltung
- + [__2 Dok.] 800 Literatur
- + [__1 Dok.] 900 Geschichte und Geografie

Abbildung 2: Klassifikationsbrowser

Laden der Personendaten

Personen und Körperschaften ("LegalEntities") sind in MILESS nicht nur einfach als Namen in den Metadaten der Dokumente eingetragen, sondern selbständige Einträge mit eigenen Metadaten wie Anschrift, EMail, Telefon usw., auf die dann in den Dokumenten verwiesen wird. Sie können nun die Beispiel-Personen laden:

```
miless:> load all entities from directory legalentities
```

oder alternativ

```
miless.cmd|miless.sh load all entities from directory legalentities
```

wobei Sie *legalentities* durch den Pfad des Verzeichnisses ersetzen, in dem die entpackten Beispiel-Personendaten liegen. Die Daten werden intern in den SQL-Tabellen `LENAMES` und `LEXML` gespeichert.

Die Datei `documentation/xml/legalEntity.xsd` enthält das W3C XML Schema für Personen- und Körperschaftsdaten.

Verzeichnisse für Suchindex konfigurieren

MILESS verwendet die Apache Lucene¹³ Suchmaschine, um in den gespeicherten Metadaten und Dateien zu suchen. Bevor Dokumente geladen werden können, müssen Sie dazu in `mycore.properties` im Abschnitt "Search Index Configuration" die Verzeichnisse konfigurieren, in denen die Suchindexdaten und Lock-Dateien von Lucene gespeichert werden. Ändern Sie dort die Konfigurationseigenschaften

```
MCR.Lucene.LockDir=/mcr/repository/lucene/locks
MIL.Documents.TextIndexDir=/mcr/repository/lucene/index
MCR.Searcher.LuceneMiless.IndexDir=/mcr/repository/lucene/index
MCR.Searcher.LuceneSCORM.IndexDir=/mcr/repository/scormindex
```

Die Verzeichnisse werden automatisch angelegt, wenn Sie noch nicht existieren. Nach Konfiguration der Verzeichnisse müssen Sie erneut

```
ant bin
```

aufrufen, damit die aktualisierte Konfigurationsdatei verwendet wird. Grundsätzlich verwenden sowohl das Command Line Interface als auch die Web-Applikation nur die Dateien im Verzeichnis `build/`, so dass laufende Umgebung und geänderte Dateien der ursprünglichen Distribution immer getrennt bleiben. Ein Aufruf von `ant bin` bzw. `ant webapp` übernimmt Änderungen an Konfigurations- und anderen Dateien in die laufende Umgebung. Änderungen sollten Sie nie in der laufenden Umgebung, d.h. nicht unterhalb des Build-Verzeichnisses vornehmen.

Laden der Dokumenten-Metadaten

Sie können nun die Metadaten (Titel, Autor usw.) der Beispiel-Dokumente laden:

```
miless:> import all documents from directory documents
```

oder alternativ

```
miless.cmd|miless.sh import all documents from directory documents
```

wobei Sie `documents` durch den Pfad des Verzeichnisses ersetzen, in dem die entpackten Beispiel-Dokumentdaten liegen. Die Daten werden intern in den SQL-Tabellen `DOCUMENTS` (Metadaten) und `DOCREL` (Verweise zwischen Dokumenten) gespeichert.

Die Datei `documentation/xml/document.xsd` enthält das W3C XML Schema für Dokumente.

¹³ <http://lucene.apache.org/>

Erkennung von Dateitypen konfigurieren

Die Datei `config/FileContentTypes.xml` definiert die dem MILESS System bekannten Dateitypen. Neben der Bezeichnung und dem MIME-Typ kann hier für jeden Dateityp ein Link zum Download eines Plug-ins angegeben werden. Weiterhin können Regeln definiert werden, über die Dateitypen anhand der Dateiendung und/oder einer Bytesequenz (Magic Bytes) erkannt werden. Beispiel:

```
<type ID="pdf">
  <label>Adobe PDF</label>
  <url>/plugins/adobe.xml</url>
  <mime>application/pdf</mime>
  <rules>
    <extension score="0.9">pdf</extension>
    <pattern score="1.0" offset="0" format="text">%PDF-</pattern>
  </rules>
</type>
```

Für jede Erkennungsregel wird ein "Score" zwischen 0 (keine Erkennung) und 1 (sichere Erkennung) vergeben. Treffen mehrere Regeln zu, werden die Werte (bis zum Maximalwert 1) addiert. Im obigen Beispiel ist also eine Datei mit Endung "pdf" mit 90 %iger Wahrscheinlichkeit eine PDF-Datei. Wenn die Datei mit der Zeichenfolge "%PDF-" beginnt, wird von einer 100 %igen Erkennung ausgegangen.

Sie können diese Liste ggf. um weitere Dateitypen ergänzen. Dateitypen werden zum Zeitpunkt des Hochladens in das System erkannt. D.h. eine Datei bisher unbekanntem Typs wird nicht nachträglich nach Hinzufügen des Dateityps in die Konfigurationsdatei erkannt; erst nach einem Update der Datei wird der Dateityp korrigiert.

Speicherorte für Dateien konfigurieren

Bevor Dateien in das MILESS System geladen werden können, müssen Sie noch konfigurieren, wo die Inhalte dieser Dateien intern gespeichert werden sollen. MILESS verwendet dazu die "FileContentStores" aus dem MyCoRe Kernsystem. Sie können Dateien in lokalen oder entfernten Dateisystemen, auf Video Streaming Servern oder im IBM Content Manager speichern. Es werden die folgenden Stores unterstützt:

- Lokales Dateisystem auf dem gleichen Server:
`MCRStoreLocalFilesystem`
- Entferntes Dateisystem, über FTP angebunden:
`MCRStoreRemoteFTP`
- IBM VideoCharger Streaming Server (Streaming von MPEG1 und MPEG2):
`MCRStoreVideoCharger`

- Helix Universal Server / Real Server
kann über einen beliebigen anderen `FileContentStore` angebunden werden
- Apache Commons Virtual Filesystem¹⁴, unterstützt SFTP, WEBDAV und CIFS
`MCRStoreVFS`
- IBM Content Manager Version 8
`MCRStoreContentManager8`

Details zu den `FileContentStores` finden Sie in der Dokumentation des MyCoRe Kernsystems¹⁵. Die Datei `mycore.properties` enthält Beispiel-Konfigurationen für die Speicherung in einem lokalen Verzeichnis und die Anbindung eines Helix Universal Streaming-Servers, was die beiden häufigsten Anwendungsfälle darstellt. In einem MILESS System werden die Metadaten der gespeicherten Dateien (MD5 Prüfsumme, Größe, erkannter Dateityp usw.) in der SQL-Tabelle `FILES` gespeichert. Der eigentliche Dateiinhalt wird in einer eigenen Struktur im `FileContentStore` gespeichert.

Für jeden `FileContentStore` sind eine eindeutige ID (z.B. `FS` oder `REAL8`), die Store-Implementierung (vgl. obige Liste) und weitere Parameter abhängig von der Implementierung (z.B. das Zielverzeichnis) angegeben. Sie müssen mindestens einen `FileContentStore` konfigurieren. Ändern Sie dazu in der Datei `mycore.properties` im Abschnitt "FS - Local Filesystem Content Store" das Verzeichnis, in dem Dateien auf dem lokalen Server gespeichert werden sollen:

```
MCR.IFS.ContentStore.FS.BaseDirectory=/mcr/repository/fs/
```

Integration eines Helix Universal Streaming Servers

Sie können Dateien auf einem RealServer 8 oder Helix Universal Server 9 (oder höhere Version) speichern und dann z.B. MP3, WAV, AVI, Real, Quicktime oder Windows Media Audio-/Video-Dateien über diesen Server streamen. Die Anbindung des Streaming Servers ist dabei für den Nutzer transparent, d.h. beim Hochladen von Dateien wird anhand des erkannten Dateityps die Datei automatisch auf dem Streaming Server gespeichert. Wenn der Nutzer die Datei später in der MILESS Weboberfläche anklickt, wird sie nun nicht heruntergeladen, sondern über den entsprechenden Player (RealPlayer oder Windows Media Player) im Streaming-Verfahren wiedergegeben.

Im Helix Server müssen Sie dazu zunächst einen neuen "Mount Point" anlegen, d.h. ein Verzeichnis, in dem die Audio-/Video-Dateien abgelegt werden sollen. Eine Anleitung dazu finden Sie im Helix Server

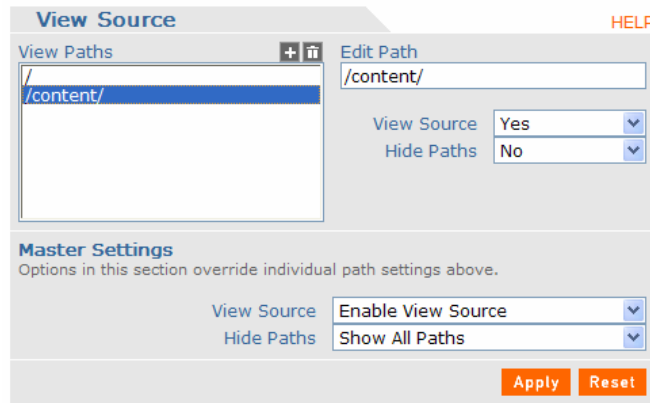
¹⁴ <http://jakarta.apache.org/commons/vfs/>

¹⁵ <http://www.mycore.de/>

Administration Guide. Legen Sie z.B. ein Verzeichnis `/mcr/repository/helix` an und mounten Sie dieses Verzeichnis im Helix-Server unter `/content/`. Sie können den Mount Point testen, indem Sie z.B. eine Datei `test.rm` in das erstellte Verzeichnis kopieren. Diese sollte dann über einen Aufruf wie

`http://server.name.de:port/ramgen/content/test.rm`

im RealPlayer wiedergegeben werden. Weiterhin müssen Sie "View Source" im Helix Server Administrator für diesen Mount Point aktivieren:



Sie können "View Source" über einen Browser testen:

`http://server.name.de:port/viewsource/template.html?src=/content/test.rm`

Nun können Sie die Anbindung des Helix Servers als Content Store in `mycore.properties` konfigurieren. Ändern Sie dazu im Abschnitt " REAL8: Helix Universal Streaming Server used as a Content Store" die folgenden Konfigurationszeilen ab:

```
MCR.IFS.ContentStore.REAL8.BaseDirectory=/mcr/repository/helix/
MCR.IFS.AVExtender.REAL8.RamGenBaseUrl=
    http://server.name.de:port/ramgen/content/
MCR.IFS.AVExtender.REAL8.AsxGenBaseUrl=
    http://server.name.de:port/asxgen/content/
MCR.IFS.AVExtender.REAL8.ViewSourceBaseUrl=
    http://server.name.de:port/viewsource/template.html?src=/content/
```

Dateiname: / MD5:	Größe der Datei:	Dateityp:	Letzte Änderung:	ZIP-Datei:
MP2005-1.rm abac0eb7572547953d7beb4fc46d6930	617,56 MB 700 kBit / Sek.	Real Video 1 Std. 33 Min. 45 Sek. mit 25 Bildern / Sek.	05.01.2006 12:26:12	als ZIP-Datei

Abbildung 3: Streaming-Video in der Detailansicht

Integration eines IBM VideoCharger Servers

Alternativ oder zusätzlich können MPEG-1, MPEG-2 oder Apple Quicktime Dateien auch über einen IBM VideoCharger Server¹⁶ gestreamt werden. Der VideoCharger Server verfügt über eine eigene, interne FTP-Schnittstelle, über die MILESS dann die in diesem Server abgelegten Dateien verwaltet. Beschreibende Daten wie Bitrate und Dauer eines Videos erhält MILESS dabei über das VideoCharger-Kommando `vslist`, das über ein CGI-Skript aufgerufen wird. Dieses Skript wird in der Datei `setup/vslist` von MILESS mitgeliefert. Sie müssen dieses Skript auf dem VideoCharger Server installieren, unter IBM AIX z. B. unter

```
/usr/lpp/avs/public/cgi-bin/vslist
```

Machen Sie dieses CGI-Skript für den Webserver ausführbar und lesbar. Testen Sie das Skript, in dem Sie z. B. für ein im VideoCharger in der Asset Group AG gespeichertes Testvideo `demo.mpg` über einen Browser die folgende URL aufrufen:

```
http://[your.videocharger.srv]/vs_public/cgi-bin/vslist?AG%20demo.mpg
```

d.h. der Name der Asset Group und der Name des Assets werden über ein Leerzeichen getrennt. Als nächstes müssen Sie in der Datei `mycore.properties` im Abschnitt "VC7: IBM VideoCharger Server used as a Content Store" die folgenden Werte anpassen, um die FTP-Schnittstelle und die zu verwendende Asset Group des VideoCharger Servers bekannt zu machen:

```
MCR.IFS.ContentStore.VC7.Hostname=videocharger.uni-essen.de
MCR.IFS.ContentStore.VC7.AssetGroup=AG
MCR.IFS.ContentStore.VC7.FTPPort=4324
MCR.IFS.ContentStore.VC7.UserID=vsadmin
MCR.IFS.ContentStore.VC7.Password=*****
```

Weiterhin müssen Sie hier die URLs angeben, unter denen die CGI-Skripte des VideoChargers erreichbar sind:

```
MCR.IFS.AVExtender.VC7.VSListURL=
  http://videocharger.uni-essen.de/vs_public/cgi-bin/vslist
MCR.IFS.AVExtender.VC7.ISCPFSelURL=
  http://videocharger.uni-essen.de/vs_public/cgi-bin/iscpfsel
```

Wie im nächsten Abschnitt beschrieben, können Sie dann den VideoCharger verwenden, um MPEG Videos auf diesem hier definierten Store "VC7" zu speichern.

¹⁶ <http://www.ibm.com/software/data/videocharger/>

Auswahlregeln für Speicherorte anpassen

In der Datei `config/ContentStoreSelectionRules.xml` können Sie die Regeln anpassen, wie die Dateien anhand des erkannten Dateityps auf die zuvor konfigurierten FileContentStores verteilt werden. Wenn keine besondere Regel zutrifft, wird der Dateiinhalt im default Store gespeichert. Wenn Sie nur einen lokalen FileContentStore eingerichtet haben, sollte der Inhalt dieser Datei also schlicht wie folgt aussehen:

```
<ContentStoreSelectionRules default="FS" />
```

Wenn Sie wie zuvor beschrieben einen Helix Universal Streaming Server oder IBM VideoCharger Server eingerichtet haben, oder andere FileContentStores verwenden wollen, können Sie angeben, welche Dateitypen in welchen Stores gespeichert werden. Das Element `type` entspricht den IDs aus der Datei `FileContentTypes.xml` (Dateitypen). Beispiel:

```
<ContentStoreSelectionRules default="FS">
  <store ID="REAL8">
    <type>mp3</type>
    <type>mpeg4</type>
    <type>wav</type>
    <type>realvid</type>
    <type>realaud</type>
    <type>wma</type>
    <type>wmv</type>
    <type>asf</type>
  </store>
</ContentStoreSelectionRules>
```

Vergessen Sie nicht, nach Änderung der Konfiguration der Dateitypen, FileContentStores und Auswahlregeln erneut

```
ant bin
```

aufzurufen, damit die aktualisierte Konfigurationsdatei verwendet wird. MyCore erlaubt es grundsätzlich auch, andere oder eigene Implementierungen für die Erkennung von Dateitypen oder für die Auswahl des FileContentStores zu verwenden, um z.B. eine Zuordnung nicht anhand des erkannten Dateitypes, sondern etwa anhand der Metadaten des zugehörigen Dokumentes zu treffen. Diese erweiterten Möglichkeiten sind in der Dokumentation des MyCoRe Internal Filesystem (IFS)¹⁷ zu finden.

¹⁷ <http://www.mycore.de/>

Beispiel-Dateien laden

Sie können nun die Dateien (PDF, HTML usw.) der Beispiel-Dokumente laden. Im MILESS Datenmodell besitzt ein Dokument ein oder mehrere Dateibereiche ("Derivate"), eine Menge von logisch zusammengehörigen Dateien. Dies kann z.B. die HTML- und die PDF-Variante des gleichen Dokumentes sein. Ein Derivat kann eine komplette Verzeichnisstruktur abbilden, d.h. etwa Unterverzeichnisse enthalten. Laden Sie die Beispiel-Derivate über

```
miless:> load all derivatives from directory derivates
```

oder alternativ

```
miless.cmd|miless.sh load all derivatives from directory derivates
```

wobei Sie *derivates* durch den Pfad des Verzeichnisses ersetzen, in dem die entpackten Beispiel-Derivate und -Dateien liegen. Die Metadaten der Dateien werden intern in der SQL-Tabelle FILES gespeichert, die eigentlichen Dateiinhalte in den zuvor konfigurierten FileContentStores.

Web-Applikation erstellen

Der Aufruf

```
ant webapp
```

erstellt eine Java Web-Applikation im Verzeichnis `build/webapp/`. Diese können Sie direkt in einem Web Application Server wie Apache Tomcat, Jetty oder IBM WebSphere einbinden. Alternativ können Sie durch Aufruf von

```
ant war
```

unter dem Namen `build/miless.war` eine Web Application Archive (*.war) Datei erstellen.

Apache Tomcat konfigurieren

Details zur Installation und Konfiguration von Apache Tomcat, sowie zur Anbindung von Web-Applikationen in Tomcat entnehmen Sie bitte der Dokumentation des Produktes¹⁸. In einer Standard-Installation von Tomcat gehen Sie z.B. wie folgt vor.

¹⁸ <http://tomcat.apache.org/>

Legen Sie im Unterverzeichnis `conf/Catalina/localhost/` der Tomcat-Installation z. B. eine Datei `miless.xml` an, mit folgendem Inhalt:

```
<?xml version='1.0' encoding='utf-8'?>
<Context docBase="/miless/build/webapp" path="">
  <Manager className="org.apache.catalina.session.StandardManager"
    pathname=""/>
</Context>
```

und ersetzen Sie den Wert von `docBase` durch das Verzeichnis, in dem Ihre `miless` Web-Applikation erstellt wurde. Wenn Sie wie im obigen Beispiel die Angabe `path=""` leer lassen, wird die Web-Applikation unter dem Kontext-Pfad `"/` gestartet, d.h. sie ist direkt unter dem Servernamen erreichbar:

`http://server.name.de:port/`

Wenn Sie beispielsweise stattdessen `path="/miless"` angeben, wird `miless` unter

`http://server.name.de:port/miless/`

aufzurufen. Weiterhin müssen Sie in der Datei `conf/server.xml` von Tomcat den Eintrag für den HTTP/1.1 Connector so ändern, dass als Encoding für eingehende Request URLs die UTF-8 Kodierung angenommen wird, z. B. ergänzen Sie dort

```
<Connector port="8080" ... URIEncoding="UTF-8" />
```

Sie können nun Tomcat starten und die Webapplikation aufrufen:



Abbildung 4: MILESS Startseite (DuEPublico Layout)

Integration von Tomcat in den Apache Webserver

Für Produktionssysteme sollte Tomcat in einen Apache Webserver integriert werden. Dies hat verschiedene Vorteile, z.B.

- Arbeitsteilung: der schnellere Apache Webserver liefert statische Dateien und Grafikdateien etc. aus, während Tomcat nur die Java Servlets ausführt.
- Root-Rechte: Tomcat als Java-Prozess muss nicht mehr mit root-Rechten gestartet werden, um unter Port 80 als Standard-HTTP-Port laufen zu können. Stattdessen wird eine Apache Webserver unter Port 80 betrieben, der die Aufrufe an einen Tomcat-Server unter anderem Port (> 1024) weiterleitet.
- Der Apache Webserver und Tomcat können auf unterschiedlichen Servern laufen.
- Apache kann eine Fehlerseite zurückliefern, wenn Tomcat nicht mehr läuft.

Für die Integration gibt es zwei verschiedene Wege, deren Vor- und Nachteile Sie der Apache- und Tomcat-Dokumentation entnehmen können:

- Verwendung des Moduls `mod_jk`
- Verwendung des Moduls `mod_proxy`

Die letztere Methode wird hier beschrieben. Dabei agiert der Apache Webserver als "Reverse Proxy" unter Verwendung des Moduls `mod_proxy`. In der Datei `httpd.conf` des Webservers geben Sie dazu an, welche URLs an den Tomcat-Server weitergeleitet werden. Beispiel-Konfiguration:

```
<VirtualHost *>
  ServerName duepublico.uni-duisburg-essen.de

  ...

  ProxyRequests Off
  ProxyVia On

  <Proxy *>
    Order deny,allow
    Allow from all
  </Proxy>

  # Proxy für Tomcat
  ProxyPass      / http://duepublico.uni-duisburg-essen.de:8180/
  ProxyPassReverse / http://duepublico.uni-duisburg-essen.de:8180/
</VirtualHost>
```

In diesem Fall läuft Tomcat unter Port 8180, und Apache wird als Reverse Proxy für alle eingehenden HTTP Requests konfiguriert. In der Tomcat-Konfiguration (Datei `server.xml`) müssen Sie dann zusätzlich vermerken,

dass Tomcat hinter diesem Reverse Proxy betrieben wird und dort die Basis-URL und Portnummer des Apache Servers angeben:

```
<Connector port="8180"  
  ...  
  proxyName="duepublico.uni-duisburg-essen.de" proxyPort="80"  
  ...  
  URIEncoding="UTF-8"  
>
```

Email-Benachrichtigung konfigurieren

Benutzer haben die Möglichkeit, in der Web-Oberfläche über dem Knopf "EMail senden" eine Email mit einem Link auf das betreffende MILESS-Dokument und einen beliebigen kurzen Begleittext zu versenden. Weiterhin verschickt MILESS beim Erstellen, Ändern oder Löschen von Dokumenten (falls so in `config/collections.xml` konfiguriert) eine Email an eine beliebige Adresse, z.B. den Administrator des Systems. Hierzu muss in der Datei `mycore.properties` im Abschnitt "Sending EMails" der verwendete ausgehende Mailserver konfiguriert werden:

```
MCR.mail.server=localhost
```

Weiterhin können Sie hier eine Vorlage für den Text der Emails bearbeiten, die bei Verwendung des oben genannten Knopfes "EMail senden" in der Dokumentenansicht versandt werden. Dabei können die Platzhalter {1} bis {5} für Absender- und Empfängeradresse, Titel und Autor des Dokumentes usw. verwendet werden, wie im folgenden Beispiel:

```
# CHANGE: Subject of EMail  
MILESS.mail.subject=DuePublico Dokumentempfehlung  
  
# The text that will appear if document has no authors  
MILESS.mail.nocreator=[Autor unbekannt / nicht angegeben]  
  
# CHANGE: The template containing the body of the email  
# {1} is "from" email address  
# {2} is document author(s)  
# {3} is document main title  
# {4} is document ID  
# {5} is optional comment from sending user  
  
MILESS.mail.template=\n\  
DuePublico - Dokumentenserver der Universit\u00e4t Duisburg-Essen\n\  
-----\n\  
{1} empfiehlt Ihnen folgendes Dokument:\n\  
\n\  
{2}:\n\  
{3}\n\  
http://duepublico.uni-due.de/servlets/DocumentServlet?id={4}\n\  
-----\n\  
{5}\n\  

```

E-Mail-Empfehlung senden	
Dokument-ID:	12636
Autor(en):	Richard von Weizsäcker
Titel:	Die EU auf dem Weg zur politischen Union?
Absender E-Mail:	<input type="text"/>
Empfänger E-Mail:	<input type="text"/> <input type="button" value="+"/> <input type="button" value="-"/>
Kopie an Absender?	<input type="checkbox"/> Bitte Kopie der E-Mail auch an Absender schicken
Nachricht:	<div style="border: 1px solid gray; height: 100px; width: 100%;"></div>
<input type="button" value="Abbrechen"/> <input type="button" value="Senden"/>	

Abbildung 5: Dokument über Email empfehlen

Zugriffsstatistiken für einzelne Dokumente

In der MILESS Dokumentansicht können Nutzer für jedes einzelne Dokument über den Knopf "Statistik" eine Zugriffsstatistik für einen bestimmten Monat abrufen, die die Anzahl Zugriffe auf die Dokumentbeschreibung, aber auch auf jede einzelne Datei in den Dateibereichen des Dokumentes auflistet. Dazu muss auf dem Server das Produkt AWStats19 installiert und konfiguriert sein. AWStats erzeugt aus den Logs des Webservers Daten-Dateien, die das MILESS StatisticsServlet dann für einzelne Dokumente filtert und aufsummiert. Ändern Sie dazu in `mycore.properties` im Abschnitt "Statistics using AWStats" den Pfad und ggf. Aufbau der Namen der AWStats Log-Dateien und tragen Sie den ersten Monat und das Jahr ein, ab dem eine Statistik für den Server in AWStats vorliegt:

```
MIL.StatisticsServlet.LogFilePattern=/miless/logs/awstatsMMYYYY.miless.txt
MIL.StatisticsServlet.MinYear=2004
MIL.StatisticsServlet.MinMonth=12
```

Details zur Konfiguration von AWStats entnehmen Sie bitte der Dokumentation des Produktes. Das obige Konfigurationsbeispiel geht davon aus, dass in der AWStats Konfiguration die Log-Dateien mit einem entsprechenden Muster angelegt wurden, z. B.

```
LogFile="/miless/logs/access-%YYYY-2-%MM-2.log"
```

¹⁹ <http://www.awstats.net/>

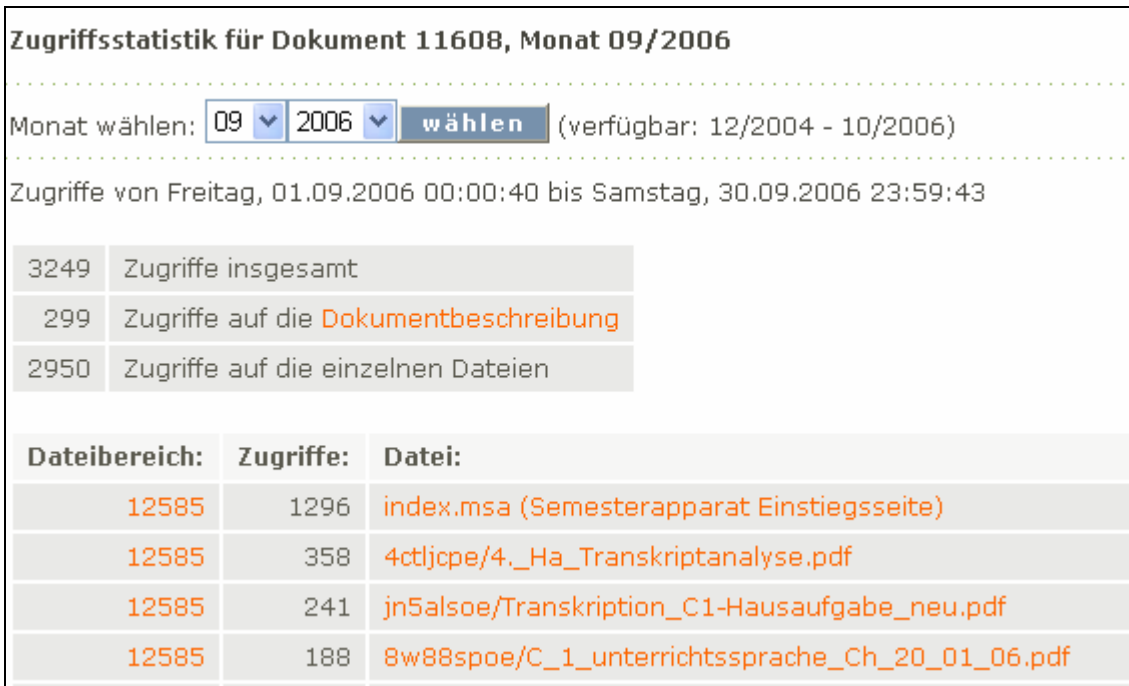


Abbildung 6: Zugriffsstatistik für ein Dokument

Personenrollen konfigurieren

Im MILESS Datenmodell verweisen Dokumente auf Personen und Körperschaften, die die Rolle von Autoren, Beitragenden oder Herausgebern spielen. Dies entspricht dem Dublin Core Datenmodell (DC.Creator, DC.Publisher, DC.Contributor). Jede dieser drei Hauptrollen kann wiederum "Unterrollen" enthalten, die die Art der Beteiligung am Dokument genauer spezifizieren. Die Datei `config/roles.xml` enthält die Bezeichnungen der möglichen Rollen, die Sie ggf. für Ihre Anwendung erweitern können. Dabei ist das Attribut "value" der Wert, der in der Darstellung der Dokumente als XML-Datei bei Import/Export und Speicherung verwendet wird, während das Attribut "label" die Bezeichnung in der Web-Oberfläche festlegt. Beispiel:

```
<roles>
  <item value="author" label="Autor">
    <item value="composer" label="Komponist" />
    <item value="painter" label="Maler" />
    <item value="programmer" label="Programmierer" />
  </item>
  <item value="publisher" label="Herausgeber / Verleger" />
  <item value="contributor" label="Beitragender">
    <item value="referee" label="Gutachter" />
    <item value="advisor" label="Betreuer/Doktorvater" />
    ...
</roles>
```

Beziehungstypen zwischen Dokumenten konfigurieren

Im MILESS Datenmodell können Dokumente in Beziehungen zu anderen Dokumenten stehen, bzw. auf andere Dokumente verweisen. Beziehungen sind immer bidirektional, d.h. können in beide Richtungen navigiert werden. In der Standard-Konfiguration von MILESS sind die folgenden drei Beziehungstypen definiert:

- ist Teil von / hat Teil
- ist neue Version von / hat neue Version
- verweist auf / wird referenziert durch

Benutzer können diese Beziehungen oder Querverweise zwischen Dokumenten in der Web-Oberfläche bei der Bearbeitung eines Dokumentes eintragen. Dies wirkt immer in beide Richtungen, d.h. wenn ein Verweis von Dokument X auf Y eingetragen wird, wird auch ein entsprechender Verweis in die Gegenrichtung von Dokument Y auf X eingetragen. Die Beziehung ist also von beiden Seiten des Verweises aus bearbeitbar oder löschtbar.

Sie können weitere Beziehungstypen bei Bedarf in der Datei `config/relations.xml` ergänzen. Für jede Beziehung muss über das Attribut `"peer"` angegeben werden, welchen Wert die Beziehung in der Gegenrichtung hat. Beispiel:

```
<relations>
  <item value="hasPart"          peer="isPartOf"
        label="ist übergeordnetes Dokument von" />
  <item value="isPartOf"        peer="hasPart"
        label="ist Teil von" />
  <item value="hasVersion"      peer="isVersionOf"
        label="hat neuere Version in" />
  <item value="isVersionOf"     peer="hasVersion"
        label="ist neue Version von" />
  <item value="references"      peer="isReferencedBy"
        label="verweist auf" />
  <item value="isReferencedBy" peer="references"
        label="wird referenziert von" />
</relations>
```

Wartungsmodus

Sie können in der MILESS Web-Applikation einen Wartungsmodus aktivieren. Während der Wartungsmodus aktiv ist, können Benutzer über die Web-Oberfläche zwar noch alle Inhalte suchen und ansehen, aber keine Änderungen vornehmen oder neue Dokumente oder Dateien einstellen. Sie können den Wartungsmodus aktivieren, in dem Sie in der Datei `mycore.properties` den Wert `MIL.System.ReadOnly` auf `true` setzen. Weiterhin können Sie eine Meldung konfigurieren, die in der Web-Oberfläche erscheint:

```
MIL.System.ReadOnly=true
```

MIL.System.ReadOnly.Message=

Aufgrund von Wartungsarbeiten ist das Erstellen oder Bearbeiten von Einträgen bis voraussichtlich ... nicht möglich.

Der Wartungsmodus eignet sich, um z. B. Korrekturen in den Datenbanktabellen, Backups oder administrative Arbeiten durchzuführen. Zur Aktivierung oder Deaktivierung des Wartungsmodus ist kein Neustart der Webapplikation erforderlich. Ändern Sie stattdessen den Wert direkt in der Datei `build/webapp/WEB-INF/classes/mycore.properties` (dies ist die Kopie der Datei `config/mycore.properties`, die zur Laufzeit der Webapplikation verwendet wird) und veranlassen Sie ein Neuladen der Konfigurationsdatei. Dies geschieht, in dem Sie irgendein MILESS Servlet mit dem zusätzlichen HTTP Request Parameter `reload.properties=true` aufrufen. Beispiel:

`http://server.name.de:port/index.xml?reload.properties=true`

Logging konfigurieren

MILESS verwendet für das Logging von Systemabläufen, Fehlermeldungen etc. das Apache Log4J Paket. In der Datei `mycore.properties` können Sie im Abschnitt "Logging with Log4J" den Log-Level anpassen und ggf. das Format und Ausgabeziel ("Appender") der Logmeldungen ändern, um sie z.B. in eine Datei zu schreiben. In der Standardkonfiguration werden die Meldungen auf der Konsole ausgegeben. Apache Log4J bietet sehr viele Möglichkeiten, das Logging individuell zu konfigurieren und unterstützt z. B. auch die Verwendung des Syslog. Details zur Konfiguration entnehmen Sie bitte der Dokumentation zu Log4J.

Für die Fehlersuche in Ihrem MILESS System können Sie den Log-Level auf "DEBUG" setzen, in dem Sie die folgenden beiden Zeilen ändern:

```
MCR.log4j.logger.miless=DEBUG,stdout
MCR.log4j.logger.org.mycore=DEBUG,stdout
```

Log4J erlaubt es, den Log-Level individuell für einzelne Java-Klassen zu steuern. Wenn Sie beispielsweise den Log-Level der Klasse `MCRURIResolver` auf DEBUG stellen, sehen Sie, welche XML Dateien und XSL Stylesheets bei der Verarbeitung einer Webanfrage nachgeladen werden. Dies ist hilfreich, um etwa die an der Erstellung einer HTML-Seite von MILESS beteiligten XSL Stylesheets zu identifizieren:

```
MCR.log4j.logger.org.mycore.common.xml.MCRURIResolver=DEBUG,stdout
MCR.log4j.additivity.org.mycore.common.xml.MCRURIResolver=false
```

Bei einem Aufruf von `DocumentServlet` (Ausgabe der Dokumenten-Titelseite) sehen Sie dann z. B. unter anderem folgende Ausgabe:

```
MCRURIResolver Including documentTrees.xsl from document.xsl
MCRURIResolver Including layout.xsl from document.xsl
MCRURIResolver Including local.xsl from layout.xsl
```

```
MCRURIResolver Including navigation.xsl from layout.xsl
MCRURIResolver Including tools.xsl from document.xsl
```

Zur Änderung der Logging-Konfiguration, z .B. um unmittelbar einen auftretenden Fehler besser analysieren zu können, ist kein Neustart der Webapplikation erforderlich. Ändern Sie stattdessen die Konfigurationswerte direkt in der Datei `build/webapp/WEB-INF/classes/mycore.properties` (dies ist die Kopie der Datei `config/mycore.properties`, die zur Laufzeit der Webapplikation verwendet wird) und veranlassen Sie ein Neuladen der Konfigurationsdatei. Dies geschieht, in dem Sie irgendein MILESS Servlet mit dem zusätzlichen HTTP Request Parameter `reload.properties=true` aufrufen. Beispiel:

```
http://server.name.de:port/index.xml?reload.properties=true
```

Datei-Upload konfigurieren

Dateien können einem MILESS Dokument auf verschiedenen Wegen hinzugefügt werden:

- Upload über ein HTML-Formular: Nur einzelne Dateien sind auswählbar. Die maximale Dateigröße ist beschränkt. Wenn ZIP-Dateien hochgeladen werden, werden diese serverseitig entpackt.
- Upload über ein Java-Applet: Erfordert, dass im Browser das Java-Plug-in von Sun installiert ist. Mehrere Dateien und auch ganze Verzeichnisbäume können komfortabel ausgewählt und hochgeladen werden. Geeignet auch für sehr viele und sehr große Dateien mit mehreren 100 MByte.

In der Datei `mycore.properties` müssen Sie für die Upload-Variante über das HTML-Formular das Verzeichnis angeben, in dem temporär die hochgeladenen Dateien zwischengespeichert werden.

```
MCR.Editor.FileUpload.TempStoragePath=/miless/uploads
MCR.Editor.FileUpload.MaxSize=10485760
```

Sie können hier die maximale Dateigröße für Uploads über das HTML-Formular angeben, z. B. 10 MByte. Unserer Erfahrung nach ist ein Upload größerer Dateien oft je nach verwendetem Browser und Rechner problematisch.

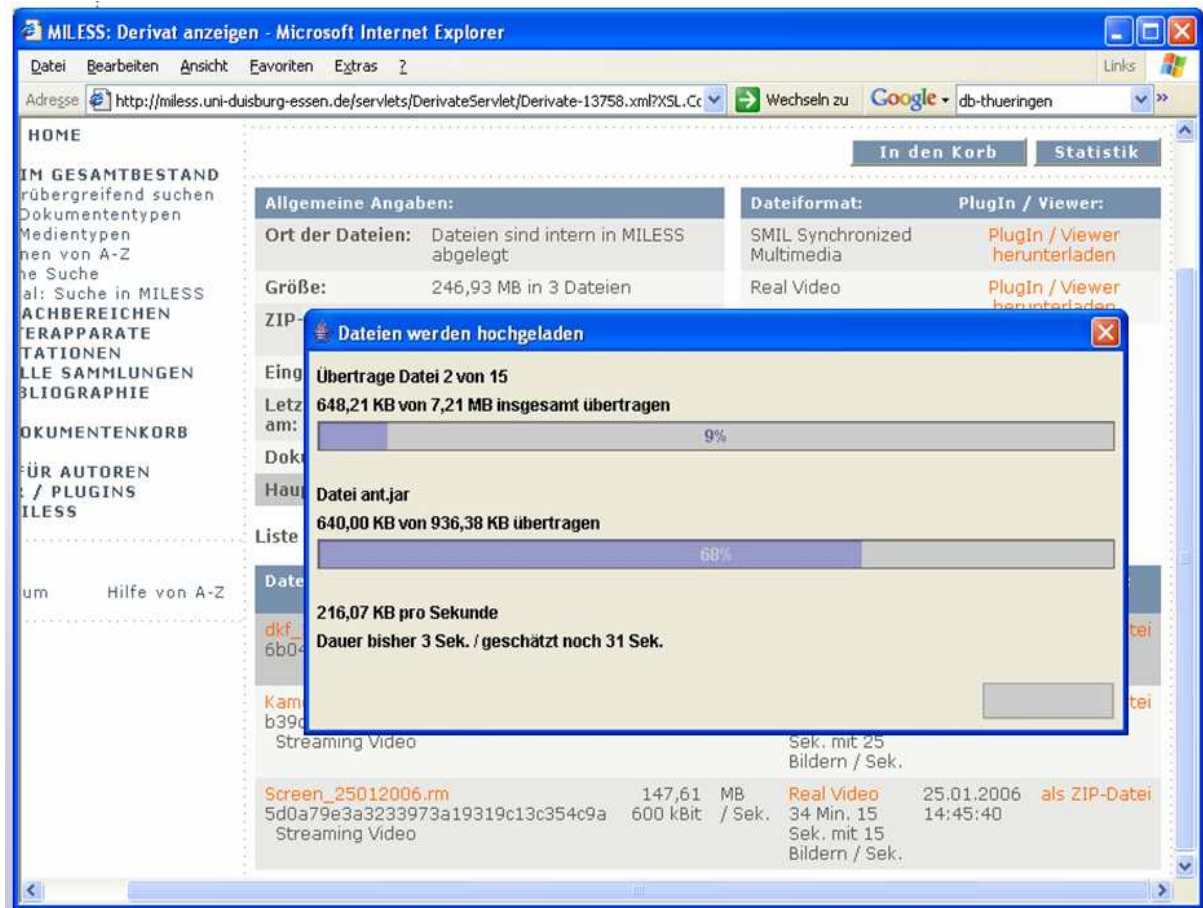


Abbildung 7: Datei-Upload mit Java-Applet

Daher sollte für große Dateien das Java-Applet verwendet werden. Dieses sendet die Dateien über eine TCP-Socket-Verbindung an den MILESS Server. Sie können die verwendete IP-Adresse des Servers und den verwendeten TCP Port konfigurieren. Falls Sie eine Firewall verwenden, muss dieser Port ggf. dort freigegeben werden. Wenn Sie keine Angaben machen, wird der Port 22471 verwendet und versucht, die IP-Adresse des Servers über den beim HTTP Request verwendeten Hostnamen des Servers zu ermitteln.

```
MCR.FileUpload.IP=132.252.123.234
MCR.FileUpload.Port=22471
```

Das Upload Applet stammt aus dem MyCoRe Kernsystem und ist als signiertes Applet in der Datei `lib/upload.jar` enthalten. Der Benutzer muss im Browser diese Signatur akzeptieren, damit das Applet berechtigt ist, Dateien von seinem lokalen Rechner zu lesen. Die Signatur hat eine begrenzte Gültigkeit. Wenn die Signatur abgelaufen ist, müssen Sie das Applet neu signieren. Ändern Sie dazu in der Datei `mycore.properties` im Abschnitt "Applet signing"

- die Angaben zu Ihrer Organisation
- die Anzahl Tage, die die Signatur gültig sein soll

- das Passwort, das bei der Erzeugung der Signatur verwendet wird
- die Datei, in der die erzeugten Signaturschlüssel gespeichert werden

```
SIGN.Alias=miless
SIGN.Name=Dokumentenserver MILESS
SIGN.OrgUnit=Universit\u00e4tsbibliothek
SIGN.Organization=Universit\u00e4t Duisburg-Essen
SIGN.Country=DE
SIGN.Validity=1000
SIGN.Password=*****
SIGN.KeyStore=/miless/keystore
```

Zunächst müssen Sie einmalig ein Schlüsselpaar für das Signieren des Applets erzeugen. Dies geschieht durch Aufruf von

```
ant genkeys
```

Die erzeugten Schlüssel werden in der unter `SIGN.Keystore` angegebenen Datei gespeichert und mit dem unter `SIGN.Password` angegebenen Passwort geschützt. Nun können Sie durch Aufruf von

```
ant applet webapp
```

die Datei `lib/upload.jar` neu erzeugen, neu signieren und in die Web-Applikation kopieren.

Download von Videos zulassen

Wenn Sie wie zuvor beschrieben einen Helix Streaming-Server oder IBM VideoCharger Server eingebunden haben, um Audio-/Video-Dateien zu streamen, können Sie einstellen, ob diese Dateien generell herunterladbar sein sollen:

- Wenn der Benutzer Schreibrechte auf das Dokument hat, oder ein Administrator ist, ist ein Download in jedem Fall erlaubt.
- Wenn der Konfigurationswert `MIL.DefaultAVStreamDownloadAllowed` auf `false` gesetzt ist, werden im Streaming Server gespeicherte Videos und Audiodateien nur gestreamt, ein Download über die MILESS Weboberfläche ist nicht möglich.
- Wenn der Konfigurationswert `MIL.DefaultAVStreamDownloadAllowed` auf `true` gesetzt ist, können im Streaming Server gespeicherte Videos und Audiodateien von allen Benutzern mit Leseberechtigung auch über den Link "ZIP erzeugen" als ZIP-Datei heruntergeladen werden.

Sie können diese allgemeine Regel auch individuell für einzelne Dateibereiche (Derivate) überschreiben, bzw. abändern. Dies geschieht durch Ändern eines Wertes in der Datenbank, momentan ist eine Änderung über die Weboberfläche nicht möglich. Wenn Sie in der Tabelle `DERIVATES` den Wert der Spalte `LOCATOR` für die entsprechende Derivat-ID auf

true oder false setzen, überschreibt dies den Wert der Konfigurationseigenschaft `MIL.DefaultAVStreamDownloadAllowed` und erlaubt bzw. verbietet explizit einen Download für den betreffenden Dateibereich. Beispiel:

```
update DERIVATES set LOCATOR='false' where ID = 4711
```

verbietet explizit einen Download von Streaming-Videos im Dateibereich mit der ID 4711.

Verteilte Suche über mehrere Server

MILESS unterstützt die Suche über mehrere, im Netz verteilte MILESS-Server. Sie können darüber z.B. parallel die Dokumentenserver in Duisburg-Essen (DuEPublico), Jena (Digitale Bibliothek Thüringen) und Ihre lokale MILESS-Installation durchsuchen. Ergänzen Sie dazu die Datei `config/hosts.xml` um einen Eintrag für Ihren lokalen Server:

```
<hosts local="due">
  <host id="due" short="Duisburg-Essen"
    label="Universität Duisburg-Essen, DuEPublico"
    base="http://duepublico.uni-duisburg-essen.de/" timeout="15000"
  />
  <host id="dbt" short="Thüringen"
    label="Digitale Bibliothek Thüringen, FSU Jena"
    base="http://www.db-thueringen.de/" timeout="15000"
  />
</hosts>
```

The screenshot shows the DuEPublico search interface. The header includes the university logo and the search title. The main content area displays search results for 'MyCoRe Entwicklertreffen' across various topics like Identity Management, Webservices, and ImageViewer. A table at the bottom summarizes the search results across different servers.

Server:	Suche:	Treffer:
Duisburg-Essen	beendet	21 / 21
Thüringen	beendet	26 / 26

Abbildung 8: Ergebnisse einer verteilten Suche

Für jeden Server muss eine eindeutige ID, eine Kurz- und Langbezeichnung, die Webadresse, sowie ein Timeout-Wert in Millisekunden angegeben werden. Wenn der angegebene Server innerhalb dieser Zeitspanne nicht reagiert, wird er bei der verteilten Suche ignoriert. Ergänzen Sie diese Datei um einen Eintrag für Ihren lokalen Server und ändern Sie das Attribut `local` auf die ID Ihres lokalen Servers.

Die Datei `web/forms/editor-distributed-search.xml` enthält eine Suchmaske für die verteilte Suche, die Sie ggf. anpassen können. Diese Suchmaske ist in die Webseite `web/search/distributed-query.xml` eingebunden.

Datensicherung und Suchindex

Sie können auf verschiedenen Wegen eine Datensicherung Ihres MILESS Systems durchführen.

- Sichern Sie die von MILESS Datenbanktabellen (je nach verwendetem Datenbanksystem z.B. über einen Export der SQL-Tabellen).
- Sichern Sie zusätzlich die Verzeichnisse auf dem Server, in denen MILESS die Dateiinhalte der Dokumente speichert (FileContentStores).

Wenn Sie ein Backup dieser beiden Bereiche erstellen, sind alle in MILESS gespeicherten Inhalte vollständig gesichert. Optional können Sie auch den Lucene Suchindex mit in die Datensicherung einbeziehen, dies ist jedoch nicht unbedingt erforderlich, da der Index jederzeit über das MILESS Command Line Interface wieder herstellbar ist. Dazu verwenden Sie ggf. das Kommando

```
miless:> rebuild document index
```

oder alternativ

```
miless.cmd|miless.sh rebuild document index
```

Dieses Kommando liest alle gespeicherten Dokumente und baut den Suchindex vollständig neu auf. Dieses Kommando ist auch zu empfehlen, um Inkonsistenzen z.B. durch einen Systemfehler oder fehlgeschlagene Update- oder Löschoperationen zu beseitigen.

Zusätzlich oder alternativ können Sie auch das MILESS Command Line Interface verwenden, das für alle gespeicherten Objekte und Dateien einen Export bzw. (Re-)Import über XML-Dateien zulässt. Die Kommandos

```
miless:> save all classifications to directory classifications
miless:> save all entities to directory legalentities
miless:> save all documents to directory documents
miless:> save all derivates to directory derivates
```

exportieren die Metadaten und Dateien aller Klassifikationen, Dokumente, Personen- und Körperschaftsdaten als XML-Dateien in lokale Verzeichnisse auf dem Server. Ersetzen Sie dabei im obigen Beispiel die Werte *classifications*, *legalentities*, *documents* und *derivates* durch die Namen leerer Verzeichnisse, in die die Inhalte exportiert werden sollen.

Für DuEPublico verwenden wir eine Kombination der oben genannten Verfahren (und zusätzlich ein "Snapshot Filesystem" auf dem zentralen Fileserver der Universität), um die Inhalte des Dokumenten- und Publikationsservers zu sichern. Sie können das Ant-Target "backup" in der Datei `build.xml` als Vorlage für Ihre eigene Datensicherungsskript verwenden und an Ihre lokale Systemumgebung anpassen, so dass Sie dann Sicherungen durch einen Aufruf von

```
ant backup
```

durchführen können.

Zugang für Internet-Suchmaschinen

MILESS besitzt mehrere Schnittstellen, um die im System gespeicherten Dokumente und Dateien auch für Internet-Suchmaschinen (Robots) wie Google zugänglich zu machen. Dazu erzeugt MILESS dynamisch einen hierarchischen, immer aktuellen Index, über den Robots zu allen Titelseiten der Dokumente gelangen (und über die Links dort auch zu den Volltext-Dateien). Dieser Index ist erreichbar unter

```
http://server.name.de:port/index/robots/index.html
```

Wenn Sie diese URL bei Suchmaschinen registrieren oder über Ihre Webseiten verlinken, können die Inhalte des Dokumentenservers von gängigen Web-Suchmaschinen gefunden und indiziert werden.

Für Google existieren noch zwei weitere mögliche Indizierungswege:

- Verwendung des Google Sitemap-Protokolls: Hierbei erzeugt MILESS dynamisch eine XML-Datei mit Links auf alle gespeicherten Dokumente. Wenn Sie sich bei Google mit Ihrem Server für das Sitemap-Protokoll registrieren, fordert Google regelmäßig diese XML-Datei vom Server an und verfolgt und indiziert die darin aufgelisteten Seiten. Die Sitemap-Datei ist erreichbar unter

```
http://server.name.de:port/sitemap.xml
```

- Verwendung der OAI-Schnittstelle: Google unterstützt ebenfalls eine Indizierung über die Schnittstelle der Open Archives Initiative. Dazu müssen Sie die Startseite Ihres OAI Data Provides bei Google registrieren:

`http://server.name.de:port/servlets/OAIDataProvider`

In jedem Fall sollten Sie andere Zugangswege zu Ihrem MILESS-System für Internet-Suchmaschinen sperren, um die Indizierungswege zu kanalisieren und eine überflüssige Belastung des Servers z. B. durch Aufruf von Webseiten, die Suchen im System auslösen, zu vermeiden. Dazu können Sie die Datei `web/robots.txt` an Ihre Bedürfnisse anpassen:

```
User-agent: *
Disallow: /servlets/NavigationServlet
Disallow: /servlets/LoginServlet
Disallow: /servlets/StoryboardServlet
Disallow: /servlets/BasketServlet
Disallow: /servlets/SearchServlet
Disallow: /servlets/DistributedQueryServlet
Disallow: /servlets/EditorServlet
Disallow: /servlets/ZipServlet
Disallow: /servlets/StatisticsServlet
Disallow: /index/names
```

Diese enthält eine Liste der URL-Muster, die für Suchmaschinen-Robots explizit nicht zugänglich sein sollen. Diese Datei muss auf der obersten Ebene Ihres Webservers zugänglich sein, z. B.

`http://server.name.de/robots.txt`

Bilder und CSS anpassen

Grundlegende Elemente des Layouts der Web-Anwendung lassen sich ohne XSL-Kenntnisse anpassen. Die folgende Tabelle gibt eine Übersicht über verwendete CSS-Dateien und Grafikverzeichnisse:

<code>web/i</code>	Grafikdateien, die Teil des Layouts der Universität Duisburg-Essen sind
<code>web/images</code>	Andere Bilder, Grafiken und Logos des Layouts
<code>web/i/unie.css</code>	CSS-Datei

Der Aufruf `ant webapp` kopiert diese Element in die laufende Web-Applikation.

XSL Stylesheets anpassen

Wenn Sie eine Webseite im MILESS System aufrufen, erzeugen die Java-Servlets als Ausgabe ein XML-Dokument. Dieses wird dann über ein XSL Stylesheet serverseitig (also nicht erst im Webbrowser des Benutzers) in eine HTML-Webseite transformiert. Wenn Sie größere Anpassungen am Layout der Webanwendung vornehmen wollen, müssen Sie diese XSL

Stylesheets anpassen. Beispiel: Die Titelseite der Dokumente wird durch das DocumentServlet erzeugt, z. B.

```
http://server.name.de:port/servlets/DocumentServlet?id=4711
```

Wenn Sie an diesen Aufruf (oder generell auch an beliebige andere Aufrufe von MILESS Servlets) den Parameter `XSL.Style=xml` anhängen, sehen Sie nicht die resultierende HTML-Webseite, sondern das ursprüngliche XML-Dokument, das das Servlet ausgegeben hat. z. B.

```
http://server.name.de:port/servlets/DocumentServlet?id=4711&XSL.Style=xml
```

```
<?xml version="1.0" encoding="UTF-8" ?>

<document ID="1" status="published" collection="LuL">
  <titles>
    <title lang="ger" translated="false" alternative="false">
      Rezension: Berthold, Siegwart: Reden lernen.
    </title>
  </titles>
  ...
</document>
```

Um die HTML-Seite zu erzeugen, wird immer zunächst ein XSL Stylesheet verwendet, das den gleichen Namen wie das Wurzelement des XML-Dokumentes trägt, also in diesem Fall `document.xml`. Die XSL Stylesheets sind im Verzeichnis `stylesheets` und Unterverzeichnissen abgelegt. Bei Zusammenstellen der Web-Applikation durch Aufruf von `ant webapp` werden sie in das Verzeichnis `build/webapp/WEB-INF/stylesheets` kopiert, von wo sie verwendet werden.

Sie können bei der Anpassung der Stylesheets diese auch direkt in diesem Verzeichnis ändern, sollten sie jedoch in das `stylesheets` Verzeichnis zurückkopieren, damit die Änderungen nicht verloren gehen. Änderungen im laufenden Betrieb werden nicht immer sofort in den Webseiten sichtbar, da MILESS die geladenen Stylesheets in einem Cache zwischenspeichert. Am einfachsten können Sie den Cache aktualisieren, in dem Sie das Änderungsdatum aller Stylesheets hochsetzen, unter Unix z. B. durch

```
touch build/webapp/WEB-INF/stylesheets/*.xml
```

Wenn Sie das Logging in MILESS anpassen, können Sie im Log sehen, welche XSL Stylesheets beim Aufruf einer Webseite an deren Entstehung beteiligt sind. Dies ist die einfachste Möglichkeit, herauszufinden, welche Stylesheets Sie bei Änderungswünschen an einer bestimmten Webseite anpassen müssen. Siehe dazu die Hinweise im Abschnitt "Logging konfigurieren".

Sie können in den MILESS XSL Stylesheet über das `xsl:param` Element auf beliebige Properties aus der Datei `mycore.properties` zugreifen, aber auch auf HTTP Request Parameter, sofern diese mit "XSL." beginnen. Dazu ein Beispiel:

`http://server.name.de:port/servlets/DocumentServlet?id=4711&XSL.foo=bar`

Wenn Sie im Stylesheet `document.xml` nun die Anweisung

```
<xsl:param name="foo" />
```

ergänzen, können Sie dort über die Variable `$foo` auf den Wert zugreifen, der beim Aufruf der Webseite übergeben wurde.

Wie oben beschrieben wird immer ein Stylesheet gesucht, das den gleichen Namen trägt wie das Wurzelement des XML-Dokumentes, das die Java Servlets zurückliefern. Sie können diesen Mechanismus jedoch erweitern und verschiedene Stylesheets für das gleiche XML-Dokument einrichten. Dabei können Sie über den Parameter `XSL.Style` im HTTP Request wählen, welches Stylesheet verwendet werden soll. Der Aufruf

`http://server.name.de:port/servlets/DocumentServlet?id=4711&XSL.Style=foo`

sorgt z.B. dafür, dass das Stylesheet `document-foo.xml` verwendet wird, d.h. der Name des Stylesheets wird aus dem Namen des Wurzelementes und dem Wert des Parameters `XSL.Style` zusammengesetzt. Sie können so z. B. einfach Varianten eines vorhandenen Stylesheets testen.

Die folgende Tabelle gibt einen Überblick über die wichtigsten Stylesheets. Alle Stylesheets sind unterhalb des Verzeichnisses `stylesheets` bzw. in Unterverzeichnissen der Module `module-*/xsl/` angeordnet.

<code>common/layout.xml</code> <code>common/local.xml</code>	Grundlegende Seitenstruktur, Layout
<code>common/navigation.xml</code>	Navigationsmenü links, Pfad oben
<code>core/login.xml</code>	Login-Seite (Benutzeranmeldung)
<code>core/webpage.xml</code> <code>common/section.xml</code>	Statische Webseiten, Suchmasken
<code>core/results.xml</code> <code>core/resultTrees.xml</code>	Trefferliste der Suche
<code>core/document.xml</code> <code>core/documentTrees.xml</code>	Detailansicht Dokument Titelseite mit den Metadaten
<code>core/derivate.xml</code> <code>core/derivateTrees.xml</code>	Detailansicht Dateibereich (Derivat): Liste der Dateien
<code>core/legalentity.xml</code> <code>core/legalentityTrees.xml</code>	Detailansicht Personen- und Körperschaftsdaten
<code>core/indexpage-legalentity.xml</code>	Personen von A-Z
<code>core/indexpage-robots.xml</code>	Zugang für Suchmaschinen
<code>core/indexpage-subselect.xml</code>	Auswahl Personen in der Eingabemaske
<code>core/items-subselect.xml</code>	Auswahl Kategorien in der Eingabemaske
<code>core/upload.xml</code>	Upload-Seite mit Formular und Applet
<code>core/abisz.xml</code>	Hilfe von A-Z, <code>web/about/abisz.xml</code>
<code>basket/*.xml</code>	Korbfunktion (Dokumente sammeln)

Anpassen der Navigationsstruktur

Die Navigationsstruktur der MILESS Webanwendung (Navigationsmenü links auf allen Webseiten) wird durch die Datei `web/navigation.xml` gesteuert. Jeder Menüpunkt entspricht einem `item` Element. Items können andere Items enthalten und so eine hierarchische Struktur bilden. Beispiel:

```
<item label="Suche im Gesamtbestand" id="gesamt"
      ref="search/search-global.xml">
  <item label="Fächerübergreifend suchen" id="search.global"
        ref="search/search-global.xml"/>
  <item label="Administratorensuche" id="search.admins"
        ref="search/search-admins.xml" allowedGroups="admins" />
  ...
</item>
```

Das Attribut `label` enthält den im Navigationsmenü angezeigten Text. Das Attribut `ref` ist ein Link auf die interne Seite, die beim Klick auf den Menüpunkt angezeigt werden soll, relativ zur Startseite der Webanwendung. Jede Webseite besitzt eine eindeutige ID. Das Attribut `id` legt fest, dass der entsprechende Menüpunkt aufklappt, wenn eine Webseite mit dieser ID angezeigt wird. Im obigen Beispiel führt ein Klick auf "Suche im Gesamtbestand" also auf die Seite `search/search-global.xml`. Wenn diese Seite angezeigt wird, hat sie die ID `search.global`, so dass sich der Menüpunkt "Fächerübergreifend suchen" öffnet. Das optionale Attribut `allowedGroups` enthält eine oder mehrere durch Leerzeichen getrennte Benutzergruppen, denen der Zugriff auf den Menüpunkt erlaubt ist. So können bestimmte Bereiche der Navigation nur bestimmten Benutzerkreisen wie Administratoren oder Autoren angezeigt werden. Mit

```
<item label="-" />
```

kann eine Trennzeile eingefügt werden. Über das Attribut `banner` kann die Bannergrafik ausgewählt werden, die im oberen Bereich des Webseitenlayouts für den aktuellen Navigationspunkt angezeigt wird. Beispiel:

```
<navigation banner="i/head.png">
  ...
  <item label="Semesterapparate" ref="semapp/index.xml"
        banner="images/banner-semapp.png">
    ...
  </item>
</navigation>
```

Das Attribut `id` der `item` Elemente bezieht sich auf die ID jeder Webseite, die MILESS ausliefert. Für statische Webseiten (Informationsseiten, Suchmasken, Eingabemasken) ist diese ID als Attribut in der Webseite enthalten, z. B. im Wurzelement der Datei `search/search-global.xml`:

```
<webpage id="search.global" print="'no'" title="Globale Suche">
```

Für die Webseiten zur hierarchischen Navigation in den Kategorien einer Klassifikation (`/servlets/NavigationServlet`, z. B. DDC) bildet sich die ID durch Anhängen der Klassifikations-ID an das Präfix "browse.", z. B. "browse.DDC"

Webseiten, Suchmasken und Eingabeformulare anpassen

Alle statischen Webseiten sind im MILESS-System als XML-Dateien vom Dokumententyp `webpage` realisiert. Die Grundstruktur einer solchen Seite baut sich wie folgt auf:

```
<webpage id="search.global" print="no" title="Globale Suche">
  <section title="Globale Suche">
    ...
  </section>
</webpage>
```

Das Attribut `id` der Webseite referenziert den Menüpunkt in der Navigationsstruktur, der sich öffnet, wenn diese Webseite angezeigt wird. Das Attribut `title` enthält den Titel der Webseite. Das optionale Attribut `print="no"` unterdrückt den Button "Seite drucken", der ansonsten für statische Webseiten erscheint und dem Nutzer einen Ausdruck der Informationen ohne Navigationsstruktur ermöglicht.

Eine Webseite enthält ein oder mehrere `section`-Elemente mit eigener Überschrift. Unterhalb des `section` Elementes kann beliebiges XHTML-Markup verwendet werden.

Alle Suchmasken und Eingabeformulare sind nicht direkt als HTML-Formulare, sondern in einer eigenen XML-Beschreibungssprache über das MyCoRe Editor Framework realisiert. Dieses erlaubt eine einfache Abbildung zwischen XML-Daten und Eingabeformularen, realisiert wiederholbare Felder und verfügt über umfangreiche integrierte Eingabevalidierungsregeln. Der MyCoRe Programmer Guide²⁰ enthält eine sehr ausführliche Dokumentation des Editor Frameworks, die Sie für größere Anpassungen zunächst lesen sollten.

Alle Suchmasken finden Sie im Unterverzeichnis `/web/search/`, die Eingabemasken im Unterverzeichnis `/web/authoring/`.

Benutzerverwaltung

MILESS unterscheidet Benutzer (bestehend aus Benutzerkennung und Passwort für das Login über die Web-Oberfläche) und Gruppen. Benutzer können Mitglieder von Gruppen sein und erhalten dadurch bestimmte Privilegien, wie z. B. das Recht, neue Dokumente einstellen zu dürfen. Lese- oder Schreibrechte auf einzelne Dokumente werden jedoch nicht

²⁰ <http://www.mycore.de/content/main/documentation.xml>

über Gruppenzugehörigkeiten geregelt. MILESS verwendet die folgenden Benutzerrollen

- **Gastnutzer** können sich alle öffentlichen Dokumente ansehen. Solange ein Nutzer nicht in der Weboberfläche mit einer eigenen Benutzerkennung angemeldet ist, ist er ein Gastnutzer mit der Benutzerkennung `gast`.
- **Lesenutzer** haben zusätzlich lesenden Zugriff auf einzelne, nicht öffentliche Dokumente. Sie werden durch Autoren oder Administratoren mit einer in der Regel nicht individuellen Benutzerkennung eingerichtet, um den lesenden Zugriff auf einzelne Dokumente zu schützen. Darüber hinaus haben Lesenutzer keine besonderen Rechte und sind keiner Benutzergruppe zugeordnet.
- **Autoren** dürfen neue Dokumente im System anlegen. Sie können sich dazu im Rahmen des Publikationsworkflows von MILESS über die Weboberfläche eine eigene Benutzerkennung einrichten und werden dann automatisch in die Gruppe `submitters` aufgenommen. Benutzer in dieser Gruppe dürfen entsprechend der Konfiguration in der Datei `config/collections.xml` neue Dokumente in den Kollektionen *E-Publikationen* und *Dissertationen* einstellen und auch nachbearbeiten, so lange diese noch nicht in den Status veröffentlicht übergegangen sind. Autoren, die durch einen Administrator zusätzlich auch in die Benutzergruppe `creators` eingetragen wurden, dürfen auch Dokumente in den anderen Kollektionen jederzeit einstellen und bearbeiten. Sie dürfen auch eigene Lesenutzer einrichten, über die sie den Zugriff auf ihre Dokumente schützen können.
- **Administratoren** dürfen alle Benutzer, Dokumente und sonstige Inhalte jederzeit ansehen, erzeugen, bearbeiten und löschen. Administratoren sind Mitglied der Benutzergruppe `admins`.

Während der Installation legt MILESS bereits die Benutzergruppen `admins`, `creators` und `submitters` an. Weiterhin werden die folgenden beiden Benutzer angelegt:

- Gastnutzer: Benutzerkennung `gast` mit Passwort `gast`.
- Administrator: Benutzerkennung `dilbert` mit Passwort `crashMyDL`

Über die Benutzerkennung `dilbert` können Sie sich bei einem neu installierten MILESS System bereits als Administrator einloggen und neue Dokumente oder Benutzer einstellen. Sie sollten zunächst das Passwort dieser Benutzerkennung über die Weboberfläche ändern.

Autoren und Administratoren können nach dem Login über den Menüpunkt **Publizieren** im linken Navigationsmenü

- neue Dokumente einstellen
- ihr eigenes Passwort ändern
- neue Lesenutzer einrichten

- die von ihnen eingerichteten Lesenutzer löschen oder deren Passwörter ändern
- zu allen Dokumenten gelangen, die sie selbst erstellt haben bzw. auf die sie explizite Schreibrechte haben

Administratoren können unter diesem Menüpunkt zusätzlich

- beliebige Benutzerkennungen anlegen, ändern oder löschen
- eine Liste aller Dokumente einsehen, zu denen explizite Lese- oder Schreibrechte eingetragen sind.

Rechteverwaltung

Zunächst regelt die Datei `config/collections.xml` im Rahmen des Publikationsworkflows ganz allgemein, welche Benutzergruppen neue Dokumente in welchen Kollektionen anlegen dürfen, und ob bzw. von welchen Benutzergruppen die Dokumente in einer bestimmten Kollektion bearbeitet oder gelöscht werden dürfen. MILESS kann darüber hinaus dann den lesenden Zugriff auf einzelne Dokumente schützen oder Bearbeitungsrechte an diesen Dokumenten an bestimmte Benutzerkennungen binden. Es werden nur die folgenden Zugriffsrechte unterschieden:

- **Leserecht:** Erlaubt Ansehen oder Download der zum Dokument gehörenden Dateien. Die beschreibenden Daten der Dokumente (Metadaten) sind dagegen immer frei einsehbar. Leserechte können für bestimmte Benutzerkennungen (z. B. Lesenutzer), aber auch für IP-Adressbereiche eingetragen werden. Ist für ein Dokument kein explizites Leserecht eingetragen, ist es öffentlich, d.h. auch Gastnutzer können die Dateien des Dokumentes ansehen. Ist ein Leserecht eingetragen, kann das Dokument nur nach Login eines Benutzers mit Leserechten bzw. nur von Rechnern im eingetragenen IP-Adressbereich aus angesehen werden.
- **Schreibrecht:** Erlaubt für ein einzelnes Dokument das Ändern, Löschen, den Upload von Dateien und das Ändern der Zugriffsrechte, sofern diese Aktionen nicht durch die Regeln in der Datei `collections.xml` verboten werden. Schreibrechte können nur für Benutzerkennungen eingetragen werden, nicht für IP-Adressbereiche. Wenn ein neues Dokument angelegt wird, wird immer ein Schreibrecht für die Benutzerkennung eingetragen, die das Dokument angelegt hat, d.h. dieser Benutzer kann nach dem Login das Dokument auch wieder bearbeiten.

Wie oben beschrieben können Leserechte für ganze IP-Adressbereiche eingetragen werden. So kann z. B. leicht der Zugang zu einem Dokument auf den Netzbereich der eigenen Universität beschränkt werden. Rechner außerhalb dieses Adressbereiches haben keinen Zugriff auf die Dateien

bzw. können erst nach Login mit einer leseberechtigten Benutzerkennung zugreifen. Um Leserechte über IP-Adressbereiche einschränken zu können, müssen Sie zunächst die Datei `config/subnets.xml` für Ihre Einrichtung anpassen. Beispiel:

```
<items>
  <item value="134.91"
    label="Universität Duisburg-Essen, Campus Duisburg" />
  <item value="132.252"
    label="Universität Duisburg-Essen, Campus Essen" />
  <item value="132.252.176"
    label="Universitätsbibliothek, Campus Essen" />
</items>
```

Zu beachten ist dabei, dass in der derzeitigen Implementierung keine Subnetzmasken, sondern nur Präfixe von IP-Adressen (also z. B. "alle Adressen beginnend mit 132.252") angegeben werden können. Dies schränkt die Nutzung auf vollständige 8-Bit-Subnetzbereiche oder einzelne IP-Adressen ein (d.h. Subnetzmasken, die ausschließlich 255 und 0 verwenden), ist aber meistens ausreichend, um z.B. den Zugriff auf ein Klasse-B-Subnetz der eigenen Universität zu beschränken.

Zugriffsrechte bearbeiten: Dokument 11608																							
Zugriffsrechte, die Sie nicht ändern dürfen:																							
Benutzer/Gruppen:	(keine Einträge)																						
Zugriffsrechte, die Sie ändern / löschen / hinzufügen können:																							
Benutzer/Gruppen:	<table border="1"> <tr> <td>DGD_Student</td> <td>darf</td> <td>ansetzen / lesen</td> <td>?</td> <td>+</td> <td>-</td> <td>↓</td> </tr> <tr> <td>SemApp</td> <td>darf</td> <td>bearbeiten / löschen</td> <td>?</td> <td>+</td> <td>-</td> <td>↓</td> <td>↑</td> </tr> <tr> <td>DGD_Dozent</td> <td>darf</td> <td>bearbeiten / löschen</td> <td>?</td> <td>+</td> <td>-</td> <td>↑</td> </tr> </table>	DGD_Student	darf	ansetzen / lesen	?	+	-	↓	SemApp	darf	bearbeiten / löschen	?	+	-	↓	↑	DGD_Dozent	darf	bearbeiten / löschen	?	+	-	↑
DGD_Student	darf	ansetzen / lesen	?	+	-	↓																	
SemApp	darf	bearbeiten / löschen	?	+	-	↓	↑																
DGD_Dozent	darf	bearbeiten / löschen	?	+	-	↑																	
Lesenden Zugriff (Dateien ansehen) beschränken auf:																							
IP-Netzbereiche:	<input type="checkbox"/> Universität Duisburg-Essen, Campus Duisburg <input type="checkbox"/> Universität Duisburg-Essen, Campus Essen <input type="checkbox"/> Universitätsbibliothek, Campus Essen																						
<input type="button" value="Abbrechen"/> <input type="button" value="Speichern"/>																							

Abbildung 9: Zugriffsrechte bearbeiten

In der Weboberfläche können die Zugriffsrechte über den Button "Zugriffsrechte" in der Dokumenten-Titelseite geändert werden. Im oberen Bereich des Formulars können Zugriffsrechte für Benutzerkennungen eingetragen werden. Im unteren Bereich können Leserechte für die zuvor definierten IP-Adressbereiche vergeben werden. Nicht jeder Benutzer mit Schreibrechten darf jedes Zugriffsrecht für Benutzer ändern (er könnte sich ansonsten z.B. selbst die Rechte

entziehen). Das Formular unterscheidet daher zwischen Rechten, die Sie ändern dürfen, und solchen, die Sie nicht ändern dürfen. Als Autor dürfen Sie z. B. die Zugriffsrechte für die von Ihnen selbst eingerichteten Lesenerutzer ändern. In der linken Spalte wählen Sie die Benutzerkennung, in der rechten Spalte das Lese- oder Schreibrecht. Über den Button "+" können Sie weitere Zeilen hinzufügen. Als Administrator können Sie alle Zugriffsrechte bearbeiten und auch alle im System vorhandenen Benutzer auswählen.

Publikationsworkflow anpassen

Jedes MILESS Dokument ist genau einer Kollektion zugeordnet. In der Standard-Konfiguration sind z. B. die vier Kollektionen Dissertationen, E-Publikationen, Semesterapparate und Lehr- und Lernmaterial/Sonstiges definiert. Eine Kollektion bestimmt,

- welche Rechte und Pflichten Autor und Betreiber des Dokumentenservers für dieses Dokument vereinbaren, z. B. ob es einer Qualitätskontrolle unterliegt, ob Langzeitarchivierung erfolgt
- welche Benutzergruppen neue Dokumente in dieser Kollektion einstellen dürfen
- ob und abhängig von welchem Dokumentenstatus die Benutzer das Dokument bearbeiten, Dateien ändern oder Zugriffsrechte anpassen dürfen
- welche Eingabemasken für Dokumentenmetadaten verwendet werden
- welche Dokumententypen in dieser Kollektion erlaubt oder verboten sind
- bei welchen Aktionen Emails an Administratoren verschickt werden

MILESS realisiert so insbesondere einen Freigabeworkflow für E-Publikationen und Dissertationen: Doktoranden können selbständig ihre Dissertation hochladen. Es wird dann eine Email an Mitarbeiter der Universitätsbibliothek verschickt. Die Dissertation ist dann zunächst im Zustand "eingereicht" (submitted). Dokumente in diesem Zustand sind für den Autor noch bearbeitbar und korrigierbar, jedoch nicht über Suchmasken (Ausnahme: Suchmaske für Administratoren) auffindbar und werden auch anderen Benutzern nicht angezeigt. Wenn ein Administrator den Status des Dokumentes auf "publiziert" (published) geändert hat, ist keine Änderung durch den Autor mehr möglich, und das Dokument wird angezeigt und ist suchbar.

Die zentrale Konfigurationsdatei für die Kollektionen ist die Datei `config/collections.xml`. Das Attribut `default` des Wurzelementes legt zunächst fest, welche Kollektion die Default-Kollektion ist, falls ein bereits vorhandenes Dokument keiner Kollektion zugeordnet ist. Es folgt eine Liste der möglichen Benutzeraktionen im Workflow:

```

<collections default="LuL">
  ...

  <actions>
    <action> create      </action> <!-- Dokument erzeugen -->
    <action> update      </action> <!-- Dokument ändern -->
    <action> delete      </action> <!-- Dokument löschen -->
    <action> files       </action> <!-- Dateien upload/ändern/löschen -->
    <action> permissions </action> <!-- Zugriffsrechte ändern -->
  </actions>

```

Die Aktionen `create/update/delete` beziehen sich auf das Erzeugen, Ändern oder Löschen der Dokumentbeschreibung (Metadaten), die Aktion `files` auf das Hochladen, Ändern oder Löschen der zum Dokument gehörenden Dateien. Der Abschnitt `accountCreation` definiert Text und Adressat einer Email. Wenn ein Benutzer sich selbst über die Weboberfläche eine neue Benutzerkennung einrichtet, wird er in die Gruppe `submitters` eingetragen, und es wird eine Email-Benachrichtigung an einen Administrator verschickt:

```

<accountCreation>
  <group> submitters </group>
  <email>
    <from>DuEPublico Server &lt;duepublico@uni-duisburg-essen.de&gt; </from>
    <to> duepublico@uni-duisburg-essen.de </to>
    <subject> Neue Benutzerkennung angelegt: %user% </subject>
    <body>
      Eine neue Benutzerkennung wurde in DuEPublico angelegt:

      Benutzerkennung: %user%

      Name:           %name%
      Fachbereich:    %origin%
      Telefon:        %phone%
      EMail:          %email%
    </body>
  </email>
</accountCreation>

```

In Text und Betreff der EMail können Platzhalter für Benutzerkennung, Name, Fachbereich, Telefon und Email-Adresse des neuen Benutzers verwendet werden. Ein weiterer, analog aufgebauter Abschnitt regelt das Verschicken einer Email, wenn ein Benutzer in eine neue Benutzergruppe aufgenommen werden möchte, um Lehr- und Lernmaterial einstellen zu dürfen.

Es folgt die Konfiguration der Kollektionen. Jede Kollektion besitzt eine eindeutige ID und eine Bezeichnung im Attribut `label`. Weiterhin kann man einschränken, welche Dokumententypen in dieser Kollektion ausschließlich erlaubt (Element `include`) oder alternativ ausdrücklich verboten sind (Element `exclude`). In der Kollektion "Dissertationen" ist z. B. nur der Dokumententyp "Dissertation" (ID `b.1` in der `TYPE` Klassifikation) erlaubt. Eine Dissertation könnte aber auch als Lehr- und Lernmaterial in einer anderen Kollektion eingestellt werden und unterliegt dort ggf. anderen Regeln:

```

<collection id="Diss" label="Dissertationen">
  ...
  <filter classification="TYPE">
    <include> b.1 </include>
  </filter>
</collection>
...
<collection id="LuL" label="Lehr- und Lernmaterial, Sonstiges" >
  ...
  <filter classification="TYPE">
    <exclude> b </exclude>
    <exclude> a.6 </exclude>
  </filter>
</collection>

```

Innerhalb einer Kollektion sind Regeln über `when`-Elemente definiert. Jede Regel schränkt explizit über die Attribute `groups` (welche Benutzergruppen), `actions` (welche Aktionen) und `status` (welcher Dokumentenstatus) einen bestimmten Anwendungsfall ein, abhängig davon, welche Aktion ausgeführt werden soll, in welchem Status sich das Dokument befindet (publiziert oder eingereicht) und welcher Benutzergruppe der derzeit angemeldete Benutzer angehört. Trifft keine der Regeln zu, ist die Aktion verboten. Trifft eine Regel zu, definiert das Element `allowed`, ob dieser Anwendungsfall erlaubt ist. Das Element `editor` definiert das für diesen Fall zu verwendende Eingabeformular, so dass abhängig vom Benutzer, Status oder Kollektion verschiedene Formulare angeboten werden können. Das Element `email` enthält ggf. Text und Adressat einer Email, die als Benachrichtigung verschickt wird, wenn dieser Anwendungsfall eintritt. Dazu einige Beispiele für die Kollektion Dissertationen:

Administratoren dürfen in dieser Kollektion neue Dokumente anlegen, sie verwenden das Eingabeformular `authoring/edit-admins.xml`:

```

<when groups="admins" actions="create">
  <allowed> true </allowed>
  <editor> authoring/edit-admins.xml?... </editor>
</when>

```

Benutzer in den Gruppen `submitters` oder `creators` dürfen ebenfalls neue Dokumente anlegen, verwenden jedoch ein anderes Eingabeformular, und es wird im Anschluss eine Email an `ediss@ub.uni-duisburg-essen.de` verschickt:

```

<when groups="submitters creators" actions="create">
  <allowed> true </allowed>
  <editor> authoring/diss-create.xml?... </editor>
  <email>
    <from>DuEPublico Server &lt;duepublico@uni-duisburg-essen.de&gt;</from>
    <to> ediss@ub.uni-duisburg-essen.de </to>
    <subject> %id% / Neue Dissertation / %author% / %title% </subject>
  </email>

```

Eine neue Dissertation wurde in DuEPublico eingestellt:

Autor: %author%

```

Titel:           %title%
Fachbereich:    %origin%
URL:           http://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=%id%

Benutzerkennung: %user%
Name:           %name%
EMail:         %email%
Telefon:       %phone%
    </body>
  </email>
</when>

```

Administratoren dürfen immer Updates der Dokumentbeschreibung durchführen, andere Benutzer nur, so lange das Dokument noch nicht veröffentlicht ist (Status `submitted`):

```

<when groups="admins" actions="update">
  <allowed> true </allowed>
  <editor> authoring/edit-admins.xml </editor>
</when>
<when groups="submitters creators" actions="update" status="submitted">
  <allowed> true </allowed>
  <editor> authoring/diss-update.xml </editor>
  ...
</when>

```

Sie können so über die Datei `collections.xml` nicht nur definieren, welche Aktionen für welche Benutzergruppen abhängig von Status und Kollektion des Dokumentes erlaubt sind, sondern auch Email-Benachrichtigungen zur weiteren Bearbeitung oder Kontrolle verschicken, und je nach Anwendungsfall unterschiedliche Eingabeformulare verwenden. Die in der Datei vorhandenen Regeln geben Ihnen eine Vorlage für eigene Anpassungen. Entsprechend diesen Regeln verwendet MILESS die folgenden Eingabeformulare, die sich z. B. im Umfang der Metadatenfelder, in der Kennzeichnung von Pflichtfeldern, in den Hilfetexten und Validierungsregeln etc. unterscheiden:

<code>authoring/edit-admins.xml</code>	Alle Eingabefelder, für Administratoren
<code>authoring/delete-document.xml</code>	Dokument löschen
<code>authoring/edit-rights-doc.xml</code>	Zugriffsrechte auf Dokument ändern
<code>authoring/diss-create.xml</code>	Neue Dissertation einstellen
<code>authoring/diss-update.xml</code>	Unveröffentlichte Dissertation korrigieren
<code>authoring/lul-create.xml</code>	Neues Lehr- und Lernmaterial einstellen
<code>authoring/lul-update.xml</code>	Lehr- und Lernmaterial ändern
<code>authoring/pub-create.xml</code>	Neue E-Publikation einstellen
<code>authoring/pub-update.xml</code>	Unveröffentlichte Publikation korrigieren

Neue Dissertation eingeben:	
(*)Titel:	Deutsch <input type="text"/> + - ↓ Englisch <input type="text"/> + - ↑ Bitte geben Sie möglichst auch einen Titel in englischer Übersetzung ein!
(*)Person(en):	Autor <input type="text"/> Auswählen Neu eingeben (bitte wählen oder eingeben) + - ↓ Gutachter <input type="text"/> Auswählen Neu eingeben (bitte wählen oder eingeben) + - ↓ ↑ Gutachter <input type="text"/> Auswählen Neu eingeben (bitte wählen oder eingeben) + - ↑ Bitte geben Sie auch den ersten und weitere Gutachter ein!
(*)Fachbereich(e):	(bitte wählen) <input type="text"/> + -
(*)Sprache:	Deutsch <input type="text"/> + -
Schlag-/Stichwörter:	<input type="text"/>
Klassifikation(en):	Auswählen (bitte auswählen: DDC, PACS, Physik-Klassifikation ...) + -
(*)Beschreibung(en) oder Abstract(s):	Deutsch <input type="text"/> + - ↓ Englisch <input type="text"/> + - ↑ Bitte geben Sie möglichst auch einen Abstract (Zusammenfassung) in englischer Übersetzung ein!
Quelle(n):	Deutsch <input type="text"/> + -
Räuml./Zeitl. Erstreckung:	Deutsch <input type="text"/> + -
Rechtliche Hinweise:	Deutsch <input type="text"/> + -
Datum (TT.MM.JJJJ):	(*)Promotion beantragt am: <input type="text"/> ? (*)Datum der Promotion: <input type="text"/> ?
(*)Veröffentlichung:	Hiermit erteile ich der Universität Duisburg-Essen das nicht-ausschliessliche Recht, dieses Dokument auf dem Dokumenten-

Abbildung 10: Formular zur Eingabe einer neue Dissertation (Auszug)

Persistent Identifier und URN-Vergabe

Die URLs von Dokumenten in einem Dokumenten- und Publikationsserver können sich im Laufe der Zeit ändern, z. B. durch Wechsel der verwendeten Software oder neue Programmversionen. Persistent Identifier sollen daher die Zitierbarkeit eines elektronischen Dokumentes gewährleisten, indem sie bei einer zentralen Instanz für das Dokument einen eindeutigen Identifier registrieren. Der Identifier verweist auf die aktuelle URL des Dokumentes im Web, die bei Bedarf geändert werden kann, ohne dass der Persistent Identifier seine Gültigkeit verliert. MILESS unterstützt derzeit explizit zwei Formen von Persistent Identifier:

- Persistent URLs (PURLs). Diese speziellen URLs werden durch einen Administrator bei <http://purl.oclc.org/> registriert und dann in den Metadaten des Dokumentes eingetragen.
- URNs (urn:nbn:de). Diese Persistent Identifier verwenden den Namensraum nbn:de ("National Bibliographic Name") des "Uniform Resource Name" (URN) Standards, der durch die Deutsche

Nationalbibliothek verwaltet wird. Details zur Funktionsweise von URNs finden Sie auf den Webseiten der DNB²¹. MILESS unterstützt die automatische Erzeugung und Zuordnung von URNs, sowie die Registrierung bei der DNB im Rahmen der Meldung von Hochschulschriften und Dissertationen dort.

Für jede Einrichtung ist bei der Deutschen Nationalbibliothek ein `urn:nbn:de` Subnamespace vorgesehen. Sie müssen zunächst in der Datei `mycore.properties` im Abschnitt "Persistent Identifier: URNs" die Regeln für die Erzeugung von neuen URNs für mindestens einen Subnamespace konfigurieren:

- Jeder Subnamespace besitzt hier eine eindeutige ID, z. B. "Essen"
- Der Algorithmus zur Generierung neuer URNs kann unter verschiedenen Java-Klassen gewählt werden.
- Je Algorithmus können weitere Parameter, z. B. ein Muster für den Aufbau der URN, konfiguriert werden.

```
MCR.URN.SubNamespace.Essen.Prefix=urn:nbn:de:hbz:465-
MCR.URN.SubNamespace.Essen.NISSBuilder=
    org.mycore.services.urn.MCRNISSBuilderDateCounter
MCR.URN.SubNamespace.Essen.NISSPattern=yyyyMMdd-HHmss-000
```

Der Algorithmus in der Java-Klasse `MCRNISSBuilderDateCounter` erzeugt neue, eindeutige URNs aus dem jeweils aktuellen Datum, der Uhrzeit und/oder einem fortlaufenden Zähler. Das `NISSPattern` gibt dabei das Muster im Format der Standard-Java-Klasse `java.text.SimpleDateFormat` vor, wobei die Ziffer 0 für den optionalen fortlaufenden Zähler steht. Die obige Konfiguration würde z. B. URNs der folgenden Form erzeugen (zusammengesetzt aus dem Subnamespace Präfix, der erzeugten NISS und der automatisch berechneten Prüfsumme):

```
urn:nbn:de:465-miless-20060622-213404-0017
urn:nbn:de:465-miless-20060622-213404-0025
urn:nbn:de:465-miless-20060622-213448-0013
```

Alternativ kann auch `MCRNISSBuilderFL` (oder eine eigene Implementierung) verwendet werden, der kürzere, aber wenig "sprechende" URNs erzeugt:

```
MCR.URN.SubNamespace.Essen.NISSBuilder=
    org.mycore.services.urn.MCRNISSBuilderFL
```

In der Weboberfläche können Administratoren in der Dokumenten-Titelseite über den Button "URN erzeugen" ein Formular aufrufen. In diesem Formular kann einer der zuvor konfigurierten Subnamespaces ausgewählt werden. Sie müssen daher in der Datei `web/authoring/createURN.xml` dieses Formular anpassen und darin die ID und Bezeichnung Ihres Subnamespaces eintragen:

²¹ <http://www.persistent-identifier.de/>

```

...
<list type="dropdown" rows="1" default="Essen">
  <item value="Essen"
    label="Campus Essen (urn:nbn:de:hbz:465-...)" />
  <item value="Duisburg"
    label="Campus Duisburg (urn:nbn:de:hbz:464-...)" />
</list>
...

```

Alternativ erlaubt das Formular auch eine freie Eingabe einer NISS, so dass URNs mit beliebigem Aufbau erzeugt werden können. Wenn eine NISS eingegeben wird, ergänzt MILESS nur die Prüfsumme. Ansonsten wird der zuvor für diesen Subnamespace konfigurierte Algorithmus verwendet, um eine neue NISS zu erzeugen. Nach Absenden des Formulars wird die neue URN in die Dokumenten-Metadaten eingetragen. Entsprechend den Regeln der Deutschen Nationalbibliothek müssen Sie diese URN nun binnen 24 Stunden dort registrieren.

Hochschulschriften an die DNB melden

MILESS unterstützt die automatische Meldung von Hochschulschriften wie z. B. Dissertationen an die Deutsche Nationalbibliothek. Dabei wird auch eine evtl. für das Dokument vergebene URN registriert. Die Meldung erfolgt per Email. Bevor Sie eine Hochschule an die DNB melden, sollten Sie eine URN erzeugt haben und das Dokument in die Dewey-Dezimal-Klassifikation (DDC) eingeordnet haben, da beides Pflichtangaben für die Meldung von Dissertationen sind. Die DDC-Einordnung ist z. B. auch Teil des Betreffs der Email, die nach Absenden des Formulars an die Adresse hs@ingest.d-nb.de geschickt wird.

Ändern Sie zunächst in der Datei `mycore.properties` im Abschnitt "Document Submission to Deutsche Nationalbibliothek" die Bezeichnung und Anschrift Ihrer Einrichtung:

```

MCR.DNB.CorporateName=Universität Duisburg-Essen
MCR.DNB.CorporateName.Address=Universitätsbibliothek Duisburg-Essen,
  Universitätsstraße 9-11, 45141 Essen, Germany

```

In der Weboberfläche sehen Administratoren bei publizierten Dissertationen einen Button "DNB-Meldung", über den sich ein Formular öffnet. Um dieses Formular anzupassen, müssen Sie zunächst in der Datei `web/authoring/mailToDNB.xml` die DNB Contact ID Ihrer Einrichtung (die Sie von der Deutschen Nationalbibliothek erhalten haben) und die Email-Adresse(n) der Bearbeiter elektronischer Dissertationen ändern. An diese Email-Adressen kann eine Kopie der DNB-Meldung verschickt werden, und sie wird als Rückantwortadresse verwendet:

```

...
<list type="dropdown" rows="1" default="L6000-0627">
  <item value="L6000-0627" label="Campus Essen (L6000-0627)" />
  <item value="L6000-0600" label="Campus Duisburg (L6000-0600)" />
</list>

```

```

...
<list type="dropdown" rows="1" default="schweers@ub.uni-duisburg-essen.de">
  <item value="asse@ub.uni-duisburg-essen.de"
    label="asse@ub.uni-duisburg-essen.de" />
  <item value="schweers@ub.uni-duisburg-essen.de"
    label="schweers@ub.uni-duisburg-essen.de" />
...
</list>
...

```

Meldung an die Deutsche Nationalbibliothek:	
D-NB ID:	Campus Essen (L6000-0627) ▾
BearbeiterIn:	schweers@ub.uni-duisburg-essen.de ▾
Zustellung:	<input checked="" type="checkbox"/> E-Mail an BearbeiterIn der UB schicken <input checked="" type="checkbox"/> E-Mail an Deutsche Nationalbibliothek schicken
Betreff:	HS/Pepperl, Tiemo/510 HS/{Name, Vorname}/{DDC-Klassifikation}
Dateibereich wählen:	14293 (DissertationTiemoPepperl.pdf) ▾
Datei mit Abstract:	(bitte ggf. Datei hier wählen) ▾
Inhalt:	<pre> <META NAME="DC.Title" CONTENT="Niederfrequenzasymptotik und Wirbelstrom- Approximation der verallgemeinerten dissipativen Maxwell-Gleichungen" /> <META NAME="DC.Creator.PersonalName" CONTENT="Pepperl, Tiemo" /> <META NAME="DC.Subject" SCHEME="DDC-Sachgruppe" CONTENT="510" /> <META NAME="DC.Subject" SCHEME="DDC" CONTENT="510" /> <META NAME="DC.Subject" SCHEME="freetext" CONTENT="Ganzraumproblem, verallgemeinerte dissipative Maxwell-Gleichungen, polynomiales und exponentielles Abklingen von Eigenlösungen, Leitfähigkeit, Elektro-Magneto- Statik, Niederfrequenzasymptotik, Eddy Currents, Wirbelstrom-Problem" /> <META NAME="DC.Description" SCHEME="freetext" CONTENT="Wir stellen in dieser </pre>
<input type="button" value="Abbrechen"/> <input type="button" value="E-Mail senden"/>	

Abbildung 11: Meldung einer Dissertation an die DNB

Im Formular müssen Sie den Dateibereich wählen, der den Volltext der Dissertation im Abgabeformat (z. B. PDF) enthält. Zusätzlich können Sie ggf. eine Datei auswählen, die einen Abstract enthält, sofern dieser als Datei und nicht als Teil der Metadaten des Dokumentes bereitgestellt wurde. Der Text der Email enthält die Metadaten der Dissertation im METADISS²² Format. Dieses Format wird durch das Stylesheet `stylesheets/core/document-dnb-mail.xsl` erzeugt und kann ggf. weiter angepasst werden.

OAI Data Provider konfigurieren

MILESS verfügt über eine Schnittstelle für das Metadata-Harvesting-Protokoll der Open Archives Initiative²³ (OAI-PMH 2.0). Die Metadaten und Volltexte von Dissertationen und anderen Publikationen können über diese Schnittstelle (OAI Data Provider) angeboten werden, werden dann durch externe OAI Service Provider gesammelt und sind damit auch auf internationaler Ebene recherchierbar.

²² <http://deposit.ddb.de/metadiss.htm>

²³ <http://www.openarchives.org/>

Zur Konfiguration eines eigenen OAI Data Providers ändern Sie zunächst in der Datei `mycore.properties` im Abschnitt " OAI-PMH 2.0 Data Provider Configuration" Name, URL und Identifier Ihres Dokumentenservers, sowie die Email-Adresse des Administrators. Außerdem müssen Sie ein Verzeichnis angeben, in dem temporäre OAI "resumption tokens" gespeichert werden können:

```
MCR.oai.repositoryname.OAIDataProvider=
    DuEPublico - University of Duisburg-Essen Publications Online
MCR.oai.repositoryidentifier.OAIDataProvider=
    duepublico.uni-duisburg-essen.de
MCR.oai.adminemail=luetzenkirchen@ub.uni-duisburg-essen.de
MCR.oai.resumptiontoken.dir=/miless/temp-oai/
MCR.oai.transferURL=
    http://server.name.de:port/servlets/ZipServlet/Derivate-%1.zip
```

OAI liefert die Datensätze gruppiert in "Sets" aus, die in etwa den Klassifikationen im MILESS System entsprechen. Sie müssen daher eine in MILESS geladene Klassifikation, z. B. DDC, angeben, deren Kategorien auf OAI Sets abgebildet werden. Jede Kategorie entspricht einem OAI Set. Alle über OAI auszuliefernden Dokumente müssen entsprechend in dieser Klassifikation eingeordnet sein:

```
MCR.oai.setscheme.OAIDataProvider=DDC
```

Evtl. werden Sie nicht alle in Ihrem Dokumenten- und Publikationsserver gespeicherten Dokumente über OAI nach außen geben wollen, sondern z. B. nur Dissertationen. Über die beiden folgenden optionalen Konfigurationseigenschaften kann eingestellt werden, dass nur solche Dokumente, die in einer bestimmten Klassifikationskategorie eingeordnet sind, über OAI bereitgestellt werden. Beispiel: Alle Dissertationen sind in der Klassifikation TYPE der Kategorie b.1 zugeordnet:

```
MCR.oai.restriction.classification.OAIDataProvider=TYPE
MCR.oai.restriction.category.OAIDataProvider=b.1
```

Die Datei `mycore.properties` enthält weiterhin Abbildungsregeln von den MILESS Klassifikationen TYPE (Dokumententypen) und FORMAT (Medientypen) auf die von DINI für OAI empfohlenen Bezeichnungen für doc-type und pub-type. Sie müssen diese Abbildungsregeln nur dann ändern, wenn Sie auch die Kategorien in diesen beiden Klassifikationen abändern, z. B.

```
# Classification "FORMAT", category "1" is "doc-type:text" at DINI
MCR.oai.dini-mapping.doc-type.text=FORMAT:1
# Classification "FORMAT", category "4" is "doc-type:image" at DINI
MCR.oai.dini-mapping.doc-type.image=FORMAT:4
...
```

Nach dem Start der Webapplikation ist Ihr lokaler OAI Data Provider nun unter der Adresse

`http://server.name.de:port/servlets/OAIDataProvider`

erreichbar. Sie können die Funktion dieser Schnittstelle über die Werkzeuge auf den Webseiten der Open Archives Initiative (Tools, OAI Repository Explorer²⁴) testen und sich dort nach erfolgreicher Validierung international als OAI Data Provider registrieren.

Web Services Schnittstellen

MILESS verfügt über Web Services Schnittstellen auf Basis des Apache Axis²⁵ Frameworks, mit denen Sie über SOAP-Nachrichten (Simple Object Access Protocol) im XML-Format aus externen Systemen heraus (z. B. Lernmanagementsystemen) Aktionen im Dokumenten- und Publikationsserver ausführen können, z. B.

- Suchen ausführen und Trefferlisten abrufen
- Dokumente, Derivate und Klassifikationen abrufen
- Dokumente und Personendaten erstellen, ändern und löschen
- Benutzer erstellen, ändern und löschen
- Zugriffsrechte abfragen und setzen
- Dateien importieren, aktualisieren und löschen

Um die Web Services Schnittstellen in Betrieb zu nehmen, ändern Sie zunächst in der Datei `module-webservices/build.xml` den Hostnamen und HTTP Port Ihres Servers, unter dem die MILESS Webapplikation läuft:

```
...  
<property name="hostname" value="localhost" />  
<property name="port" value="8080" />  
...
```

Dann führen Sie die folgenden Schritte aus:

- Starten Sie die MILESS Webapplikation
- Wechseln Sie in das Verzeichnis `module-webservices`
- Rufen Sie in der Kommandozeile `ant deploy` auf
- Zum Deaktivieren der Webservices rufen Sie ggf. `ant undeploy` auf

Die Web Services sind nun in die Webapplikation integriert und können benutzt werden. Als ersten Funktionstest rufen Sie

`http://server.name.de:port/servlet/AxisServlet`

auf. Sie erhalten darauf hin eine Liste der aktivierten Web Services mit deren Operationen. Für jeden Web Service können Sie über den dort

²⁴ <http://re.cs.uct.ac.za/>

²⁵ <http://ws.apache.org/axis/>

gezeigten Link die WSDL-Beschreibung (Web Services Description Language) der Schnittstelle abrufen, z. B.

```
http://server.name.de:port/services/QueryService?wsdl
```

Sie können nun eine Methode des Web Service über eine SOAP Nachricht im XML-Format entsprechend der WSDL-Spezifikation aufrufen. Alternativ können Sie für erste Tests auch die meisten Web Services über einfach HTTP GET Requests (URLs mit Parametern) aufrufen. So können Sie z. B. über den Aufruf

```
http://server.name.de:port/services/QueryService?  
method=doQuery&query=MyCoRe
```

eine Suche nach allen Dokumenten ausführen, die den Begriff MyCoRe in den Metadaten enthalten und erhalten als Antwort eine Liste der IDs aller Dokumente, auf die die Suchanfrage zutrifft. Über die Web Services Schnittstellen können Sie sehr komplexe Suchanfragen inklusive Verwendung boole'scher Operatoren formulieren und die Ergebnisse in verschiedenen Formaten und Detailgraden abrufen. Weitere Details zur Verwendung der Suchschnittstelle entnehmen Sie bitte der Dokumentation in der Datei `module-webservices/webservices.txt`.

Die weiteren Web Services ermöglichen auch das Erzeugen, Ändern, Retrieval und Löschen von Dokumenten, Personen, Derivaten, Dateien Benutzern und Zugriffsrechten bis hin zum Upload von Dateien aus einem externen System. Weitere Informationen dazu finden Sie

- in den WSDL-Spezifikationen der Web Services
- in der JavaDoc Quellcode-Dokumentation des Pakets `miless.webservices`

Bitte beachten Sie, dass sich die Web Services Schnittstellen noch in Weiterentwicklung befinden und in der aktuellen Implementierung ungeschützt sind. Sie sollten also mit anderen Mitteln (z. B. durch einen Zugriffsschutz über den Apache Webserver) dafür sorgen, dass nur autorisierte Drittsysteme Zugang zu allen URLs unterhalb von `http://server.name.de:port/services/` bekommen.

SCORM-Lernmaterial suchen und anzeigen

MILESS unterstützt E-Learning-Material, das nach dem SCORM (Shareable Content Object Reference Model) Content Packaging Standard zusammengestellt ist. Ein SCORM-Paket ist eine Zip-Datei, die eine Datei `imsmanifest.xml` mit Struktur- und Metadaten enthält.

- **Import:** Sie können über Web Services oder die Weboberfläche SCORM-Pakete einfach als Dateibereiche eines MILESS Dokumentes

laden. So können Sie E-Learning-Kursinhalte im Dokumenten- und Publikationsserver archivieren und verteilen.

- **Export:** Über die in die Weboberfläche integrierte ZIP-Funktion können Sie SCORM-Dateibereiche wieder als ZIP-Paket exportieren und wieder in Ihr Lernmanagementsystem überführen.
- **Viewer:** Über den integrierten SCORM Viewer können Sie Lernmaterial direkt im Dokumentenserver ansehen und dabei auch die Navigationsstruktur des Paketes nutzen. Wenn Sie in MILESS eine `imsmanifest.xml` Datei anklicken, wird der Viewer automatisch gestartet.
- **Suche:** Sie können in den E-Learning SCORM- und LOM-Metadaten aller Pakete qualifiziert suchen und so für Ihre Lehrveranstaltung relevantes Material finden. Parallel können Sie auch die Volltextsuche in den Inhalten verwenden. Dateibereiche, die SCORM `imsmanifest.xml` Dateien enthalten, werden automatisch für die SCORM Suche indiziert.

Um die Suche in SCORM-Paketen zu ermöglichen, müssen Sie in der Datei `mycore.properties` ein Verzeichnis angeben, in dem der Suchindex gespeichert wird, z. B.

```
MCR.Searcher.LuceneSCORM.IndexDir=/miless/scormindex
```

Eine prototypische Suchmaske für SCORM- und LOM-Metadaten finden Sie in der Datei `module-scorm/web/edit-searchScorm.xml`. Zusammen mit der Datei `config/searchfields.xml` legt sie die Metadatenfelder fest, in denen Sie suchen können. Die Beispielkonfiguration ermöglicht die Suche in den Titel-Feldern aller SCORM Item-Elemente und den LOM Description-Elementen dieser Items. Wenn Sie die Suchmaske verwenden, finden Sie die Dokumente, die der Anfrage entsprechende SCORM Metadaten enthalten. Jeder Treffer ist mit Links versehen, die Sie unmittelbar zu den entsprechenden SCORM Items in der Paketstruktur führen.

Die Suche in SCORM- und LOM-Metadaten ist bereits wie beschrieben nutzbar, aber noch prototypisch implementiert und wird zur Zeit von uns weiter ausgebaut.

Semesterapparate online

Online-Semesterapparate sind in MILESS als spezieller Dokumententyp verfügbar. Ein Online-Semesterapparat enthält

- eine Liste der im Semesterapparat enthaltenen Bücher mit bibliographischen Angaben und Link in den Bibliothekskatalog.
- Zeitschriftenartikel und Buchauszüge mit bibliographischen Angaben und Volltext im PDF-Format
- WWW-Links mit Kommentaren

- Verweise auf andere Dokumente in MILESS, z. B. Vorlesungsskripte/-folien oder Audio-/Video-Material
- Weitere Dateien, z. B. Übungsblätter oder digitalisierte Kopiervorlagen
- beliebige HTML-Abschnitte, die die Dozenten über einen WYSIWYG-Editor komfortabel online bearbeiten können.

Die Zeit der Staufer

Dr. Runde, Ingo - Semesterapparat Nr. 65 (Essen, Lesesaal GW/GSW) - DuEPublico Dokument 13636

Inhaltsverzeichnis

- [Literatur](#)
- [Quelleneditionen](#)
- [Aufsatzrecherche](#)
- [Handout-Download](#)

Literatur

↓ X

↑ ▲ **Stauferzeit und spätes Mittelalter / Hartmut Boockmann**
 Sonderausg., für die Sammlung Siedler durchges. u. auf den neuen Stand gebrachte Ausg. - Berlin : Siedler, 1994. - ISBN 3-88680-500-X
 Signatur: [LKCC1725-4_d \(Im Bibliothekskatalog anzeigen...\)](#)

↑ ▲ **Zwölftes Jahrhundert / Alfred Haverkamp**
 Stuttgart : Klett-Cotta, 2003. - ISBN 3-608-60005-1
 Signatur: [LKC1035\(10\)-5 \(Im Bibliothekskatalog anzeigen...\)](#)

↑ ▲ **Die Staufer / Odilo Engels**
 7., verb. und erg. Aufl. - Stuttgart [u.a.] : Kohlhammer, 1998. - ISBN 3-17-015157-6
 Signatur: [LLH1076\(7\) \(Im Bibliothekskatalog anzeigen...\)](#)

Abbildung 12: Online-Bearbeitung eines Semesterapparates

Dozenten bearbeiten jederzeit alle Einträge ihrer Semesterapparate online über einfache Eingabe-Formulare in MILESS. Mit der Bearbeitungsfunktion ist ein Benachrichtigungsmechanismus über Emails verknüpft, über den Mitarbeiter der Universitätsbibliothek z. B. beauftragt werden, einen Zeitschriftenartikel zu digitalisieren und als PDF einzustellen. Das System verfügt weiterhin über eine Anbindung an das Aleph Bibliothekssystem, über das sich Literaturlisten mit dem Präsenz-Semesterapparat synchronisieren und bibliographische Angaben übernehmen lassen. Online-Semesterapparate auf MILESS Basis werden neben der Universität Duisburg-Essen (im Sommersemester 2006 über 600 Semesterapparate mit mehr als 8000 Volltexten) inzwischen auch an den Universitäten Paderborn und Düsseldorf eingesetzt.

Die Einträge der Online-Semesterapparate sind in XML-Dateien mit der Bezeichnung `index.msa` enthalten. Alle zum Bereich Online-Semesterapparate gehörenden Webseiten, Konfigurationsdateien, Stylesheets und Formulare sind im Verzeichnis `module-semapp` zusammengefasst. Das Dokument `module-semapp/doc/SemesterapparateOnline.doc` enthält Dokumentation zur Implementierung und Konfiguration der Semesterapparate, die derzeit weiter vervollständigt wird. Wir empfehlen Ihnen jedoch, bei Interesse an einer Nachnutzung sehr gern mit der Universitätsbibliothek Duisburg-Essen Kontakt aufzunehmen.