

# MIR

2016.06 2017.06  
2016-07-19

by Kathleen Neumann

## Table of contents

|   |   |
|---|---|
| 1 Installation.....                           | 2 |
| 2 Konfiguration.....                          | 6 |
| 3 Anpassungen.....                            | 7 |
| 3.1 Aktivierung der Selbst-Registrierung..... | 8 |
| 4 Aktualisierung.....                         | 9 |



## 1 Installation

---

### Voraussetzungen

Neben den allgemein für MyCoRe-Anwendungen geltenden [Systemanforderungen](#) ( [../documentation/getting\\_started/requirements.html](#)) benötigen Sie für den Betrieb einer MIR-Anwendung noch die folgende Software:

- `bibutils`

### Download

Informationen zum aktuellen Release und ein Changelog befinden sich im [Download-Bereich](#) ( [../download/mir/release\\_2018.html](#)) . Bei Sonatype kann die aktuelle [MIR-Version](#) ( <https://oss.sonatype.org/service/local/artifact/maven/content?r=snapshots&g=org.mycore.mir&a=mir-webapp&v=LATEST&e=war>) (speichern als `mir.war`) heruntergeladen werden. Dieses kann mit dem Context 'mir' direkt in einen Servlet Container ausgeliefert (deployed) werden.

Für administrative Zwecke kann zusätzlich eine MIR-CLI in zwei verschiedenen Formaten heruntergeladen werden:

1. [ZIP archive](#) ( <https://oss.sonatype.org/service/local/artifact/maven/content?r=snapshots&g=org.mycore.mir&a=mir-cli&v=LATEST&e=zip>) (speichern als `mir-cli.zip`)
2. [tar.gz archive](#) ( <https://oss.sonatype.org/service/local/artifact/maven/content?r=snapshots&g=org.mycore.mir&a=mir-cli&v=LATEST&e=tar.gz>) (speichern als `mir-cli.tar.gz`)

### Solr-Installation

Soll ein eigener Solr-Server aufgesetzt werden, sind weitere Informationen dazu in unserer [Solr-Dokumentation](#) ( [../documentation/./archiv/version2017/solr\\_4.html](#)) zu finden. Den Solr-Kern für MIR liefert die Installation über den MIR-Wizard mit. Soll nicht das vom MIR-Wizard angelegte `solr-home` verwendet werden, so muss das Verzeichnis `mir/data/solr/collection1` entsprechend umkopiert werden. Auch der Name des Solr-Kerns kann beliebig angepasst werden. Wo der passende Kern gefunden werden kann, wird im MIR-Wizard konfiguriert.

Weitere Möglichkeiten der Konfiguration und mehr Informationen zur Solr-Installation finden sich in der [Solr-Dokumentation](#) ( <http://wiki.apache.org/solr/SolrInstall>) .

### Datenbank

MIR bringt H2 und HSQLDB als im-Speicher-laufende relationale Datenbanksysteme mit. Bei der Installation können diese beiden (neben verschiedenen anderen) gewählt werden. Dann ist nichts weiter zu tun. Soll eine andere Datenbank genutzt werden, so muss diese entsprechend im Wizard konfiguriert werden und ggf. weitere Anpassungen in der durch den Wizard angelegten `hibernate.cfg.xml` vorgenommen werden. Das zur Datenbank passende Treiber-Paket wird im `lib`-Verzeichnis abgelegt und muss ggf. ersetzt werden, sollte der Wizard kein passendes Paket finden können. Weitere

Information zur Konfiguration sind auf in der Dokumentation zu [Datenbank & Hibernate](#) ( ../documentation/ ../archiv/version2015/hibernate.html) zusammengetragen.

bibutils

bibutils ist ein kleines Kommandozeilenwerkzeug, das zwischen verschiedenen gängigen Bibliotheksformaten konvertiert. Um die Exportschnittstelle bei MIR vollständig zu unterstützen, muss aus der Webanwendung heraus dieses Kommandozeilenwerkzeug ausgeführt werden können. Der Nutzer mit dem der ServletContainer läuft muss also beispielsweise den Befehl `bib2xml` ausführen können (konvertiert BibTeX nach MODS-XML).

Dokumentation, Download und Installationsanleitung zu den bibutils sind auf der [Sourceforge-Projekt-Seite](#) ( <http://sourceforge.net/projects/bibutils/>) zu finden. Für Windows kann eine veraltete Version (3.4, aktuell ist 5.6 - Stand 08/2015) auf der [ehemaligen bibutils-Homepage](#) ( <http://bibutils.refbase.org/>) heruntergeladen werden.

MIR-Wizard

Nachdem das `mir.war` durch den ServletContainer bereitgestellt (deployed) wurde, erreicht man über die entsprechende Anwendungs-URL den MIR-Wizard, mit dem die Installation abgeschlossen wird. Wird Tomcat mit den Standard-Einstellungen genutzt, wäre das beispielsweise die URL <http://localhost:8080/mir/>.

Nach Aufruf dieser URL wird der MIR-Wizard gestartet, der wie Abb. 1 zeigt, ein Sicherheits-Token abfragt, das sich im Log des Servletcontainers (z.B. Tomcat-Log, siehe Abb. 2) befindet. Sucht man im Log nach "Login token", findet man die entsprechende Stelle recht schnell. Das Token - eine UUID - muss dann vollständig kopiert und eingefügt werden. Danach kann die Installation beginnen.



The screenshot shows a web form titled "Anmeldung mit Token". A light blue instruction box says: "Bitte kopieren Sie den Anmelde-Token aus der Log-Datei und tragen Sie diesen in das Feld 'Token' ein." Below this is a text input field labeled "Token" containing the UUID "ec5fea14-65e7-41bb-b9dc-07663095a38f". A blue "Anmelden" button is at the bottom right. A handwritten note in black ink says "Token aus Log kopieren und hier einfügen" with a blue arrow pointing to the token input field.

**Abbildung 1:** MIR-Wizard - Sicherheitsabfrage nach einem Token

```
[INFO] [talledLocalContainer] INFO [main] (MIRWizardStartupHandler.java:107) - Register MIRWizardRequestFilter...
[INFO] [talledLocalContainer] INFO [main] (MIRWizardStartupHandler.java:116) - Register MIRWizardServlet...
[INFO] [talledLocalContainer] INFO [main] (MIRWizardStartupHandler.java:168) -
[INFO] [talledLocalContainer]
[INFO] [talledLocalContainer] =====
[INFO] [talledLocalContainer] MIR Wizard
[INFO] [talledLocalContainer] =====
[INFO] [talledLocalContainer] Login token: 322b7d31-3a5d-4383-8684-6aa6561f6ed5
[INFO] [talledLocalContainer]
[INFO] [talledLocalContainer] =====
[INFO] [talledLocalContainer] INFO [main] (MCRStartupHandler.java:103) - 2147483647: Starting MCRFileSystemPromoter
[INFO] [talledLocalContainer] INFO [main] (MCRFileSystemPromoter.java:99) - Promoting filesystem ifs: org.mycore.datamode
```

### Abbildung 2: MIR-Wizard - Token-Logausgabe auf der Shell

Auf der folgenden Konfigurationsseite des Wizards (Abb. 3) muss nun bekannt gegeben werden, wo der Solr-Kern für die MIR-Anwendung zu finden ist. Läuft Solr im gleichen ServletContainer wie die MIR-Anwendung und wird die Default-Einstellung verwendet, muss hier nur ggf. der Port angepasst werden (z.B. auf 8080 bei Standard-Tomcat-Installationen). Die SMTP-Konfiguration ist optional, wird diese leer gelassen, kann die Anwendung keine Mails versenden. Mit funktionierender SMTP-Konfiguration versendet die Anwendung Mails bei der Selbstregistrierung um die Mail-Adresse zu validieren und auch im Rahmen des Publikationsworkflows. Diese Konfiguration kann auch nachträglich erfolgen. Als Datenbank kann zu Testzwecken ersteinmal H2 oder HSQLDB ausgewählt werden. Für den produktiven Betrieb empfehlen wir jedoch keine imRAM-Datenbank zu nutzen.

**SOLR Server**

URL:

**SMTP Einstellungen**

Host:

Anzahl der Versuche:

Benutzername:

Passwort:

**Datenbank**

Typ:

URL:

Benutzername:

Passwort:

zusätzliche Optionen:

**Abbildung 3:** MIR-Wizard - Konfiguration der MIR-Anwendung

Anschließend auf "Speichern" drücken. Nun werden Konfigurationsdateien erzeugt, Datenbanktreiber heruntergeladen, ein Solr-Home mit einem Solr-Kern für die MIR-Anwendung angelegt und Klassifikationen, Nutzer und Rechedaten geladen. Sind alle Schritte auf der nachfolgenden Seite erfolgreich abgeschlossen worden, muss als nächstes der ServletContainer einmal neu gestartet werden. Dazu kann man direkt den Knopf "Server Herunterfahren" am Ende der Seite verwenden. Wer nicht den gesamten ServletContainer neustarten möchte, kann auch nur die MIR-Anwendung neustarten. Erst mit diesem Neustart werden die neuen Konfigurationen gefunden und eingelesen. Dann steht unter <http://localhost:8080/mir/> die Anwendung zur Verfügung und man kann sich mit dem Standardnutzer: "administrator" und dem Standardpasswort: "alleswirdgut" anmelden.

Im Rahmen der MIR-Installation wurde ein Konfigurationsverzeichnis angelegt. Dieses befindet sich typischer Weise unter:

- Windows: c:\Users\

- Unix: ~/ .mycore/mir

... wobei mir beispielhaft für den Context-Namen der Webanwendung steht. Diese Anleitung bezieht sich auf Tomcat als ServletContainer mit mir und solr als Anwendungen direkt unter localhost:8080. Sollte die eigene Umgebung davon abweichen, muss die Konfiguration entsprechend angepasst werden.

```
"use strict"; (function($) { $(document).ready(function() { //Chrome support if (typeof String.prototype.endsWith !== 'function') { String.prototype.endsWith = function(suffix) { return this.indexOf(suffix, this.length - suffix.length) !== -1; }; } $.fn.resolveArtifacts = function(version) { this.each(function() { var element = $(this); var artifact = element.attr('data-artifact'); var suffix = element.attr('data-suffix'); $.getJSON('http://artifactory.mycore.de/api/search/gavc', { "g" : "org.mycore.mir", "a" : artifact, "v" : version }, function(data, textStatus, jqXHR) { var results = data.results; var uris = []; for (var i = results.length - 1; i >= 0; i--) { var e = results[i]; if (e.uri.endsWith(suffix)) { uris.push(e.uri); } } uris.sort().reverse(); $.getJSON(uris[0], function(data, textStatus, jqXHR) { element.attr('href', data.downloadUri); }); }); }); return this; }); var version = "2017.07-SNAPSHOT"; $('[data-role=artifactResolver]').resolveArtifacts(version); }); (jQuery);
```

## 2 Konfiguration

Aufbau des Konfigurationsverzeichnis

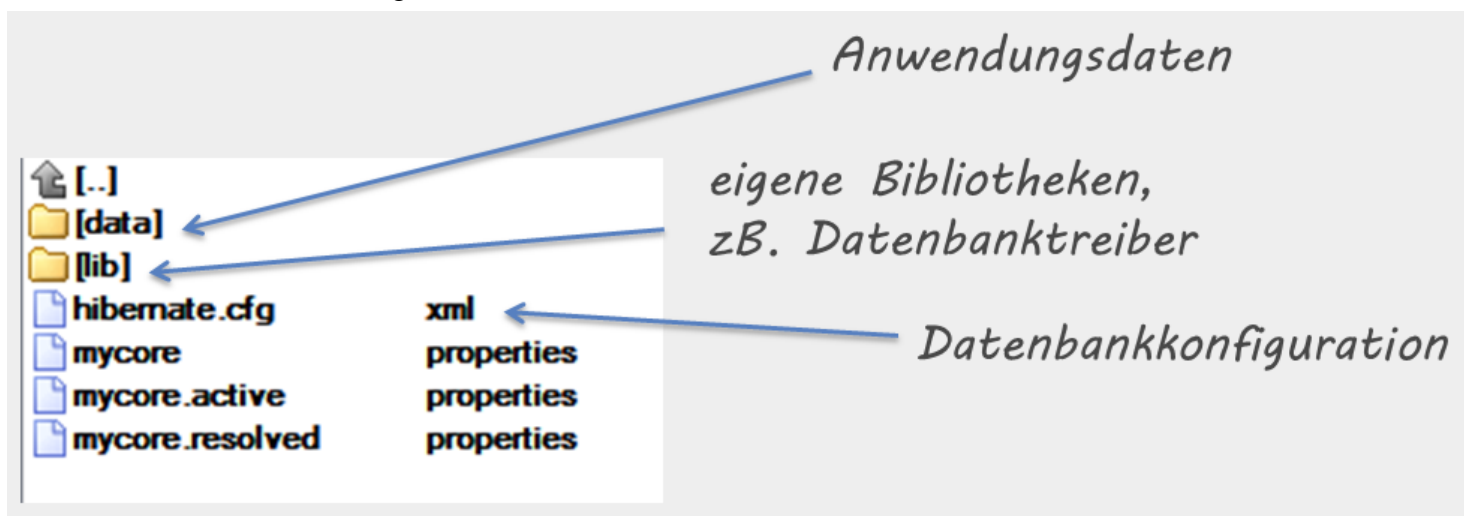


Abbildung 4: MIR-Konfigurationsverzeichnis Properties

### **mycore.properties**

enthält alle eigenen Properties, z.B. die URL zum verwendeten Solr-Core

### **mycore.active (nicht bearbeiten!)**

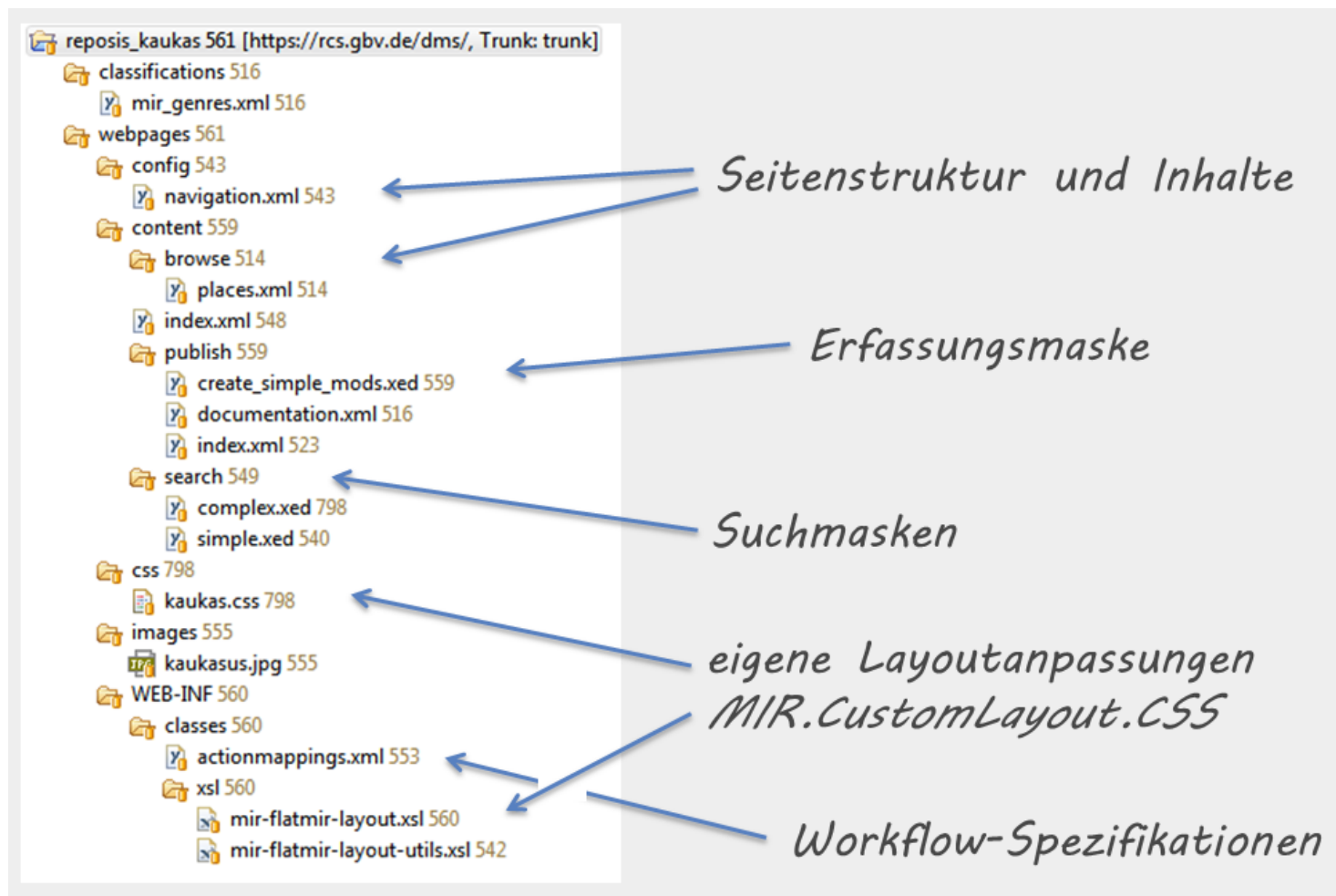
listet alle verfügbaren (aktiven) Properties inkl. Kommentaren dazu und kann somit als Vorlage zur Übernahme in die eigenen Properties dienen

### **mycore.resolved (nicht bearbeiten!)**

die aufgelösten Properties, wie sie in der laufenden Anwendung genutzt werden

### 3 Anpassungen

Alle Anpassungen am Layout, Webseiten-Inhalten, Erfassungsmasken etc. werden im Verzeichnis `%MCR.datadir%/save/webpages` hinterlegt. Beim Starten des Servlet-Containers wird der Inhalt dieses Verzeichnisses über das ausgepackte webapp-Verzeichnis im Servlet-Container kopiert und somit die default-Inhalte des `mir.war` überschrieben. Die nachstehende Abbildung gibt einen Überblick über typische Anpassungen bei einer eigenen MIR-Anwendung.



**Abbildung 5:** MIR-Webseitenverzeichnis

Als Vorlage können dafür die Dateien in den MIR-Komponenten verwendet werden. Dateien, die an der gleichen Stelle liegen (unterhalb von `webpages` == unterhalb von `resources` in den MIR-Komponenten) und den gleichen Namen haben wie in einer MIR-Komponente werden dann durch die eigene Angabe überschrieben. Alle Inhaltsseiten, Erfassungsmasken und Stylesheets zur MIR-Anwendung liegen in [mir-module/src/main/resources/](http://www.mycore.de/viewvc/viewvc.cgi/maven/mir/trunk/mir-module/src/main/resources/) (<http://www.mycore.de/viewvc/viewvc.cgi/maven/mir/trunk/mir-module/src/main/resources/>). Für Layoutanpassungen kann eine eigene `css-`

Datei hinterlegt werden, die im Verzeichnis `css` wie in der obigen Abbildung dargestellt abzulegen ist. Sollte das Basis-HTML angepasst werden sollen, so stehen hierfür die beiden XSL-Dateien `mir-flatmir-layout.xsl` und `mir-flatmir-layout-utils.xsl` in [mir-flatmir-layout/src/main/resources/xsl/](http://www.mycore.de/viewvc/viewvc.cgi/maven/mir/trunk/mir-flatmir-layout/src/main/resources/xsl/) (<http://www.mycore.de/viewvc/viewvc.cgi/maven/mir/trunk/mir-flatmir-layout/src/main/resources/xsl/>) zur Verfügung.

Sollten diese Möglichkeiten zur Layoutanpassung nicht genügen, so empfehlen wir (auch in Hinblick auf Migrierbarkeit), ein eigenes Layoutmodul analog zum `mir-flatmir-layout` anzulegen. Diese jar-Datei kann dann im `mir/libs`-Verzeichnis bereitgestellt und über das Property `MCR.LayoutTransformerFactory.Default.Stylesheets` eingebunden werden.

`MCR.LayoutTransformerFactory.Default.Stylesheets=xsl/mir-myapp-layout.xml`

### 3.1 Aktivierung der Selbst-Registrierung

Um als Alternative zur Anmeldung auch eine Selbstregistrierung anzubieten, kann dies in der `realms.xml` im Verzeichnis `data` konfiguriert werden. Dazu muss der nachstehende Code ergänzt werden (siehe dazu auch die [hier verfügbare vollständige realms.xml](#)). (`/downloads/mir/realms.xml`)

```
<realm id="registerUser" setable="false"> <label xml:lang="de">Registrierung</label>
<label xml:lang="en">Register</label> <login url="./authorization/new-author.xed"
redirectParameter="url"> <label xml:lang="de">Neue Benutzerkennung anlegen</label>
<label xml:lang="en">Create new User ID</label> <info> <label xml:lang="de"> Neuen
Benutzer für die Anwendung registrieren. </label> <label xml:lang="en"> Register new
user for application. </label> </info> </login> </realm>
```

MIR bringt bereits entsprechende Formulare und Logik mit. Um Spam zu vermeiden nutzen diese das Google-Captcha, für das entsprechend ein Schlüsselpaar registriert und in der MyCoRe-Konfiguration eingetragen werden muss. Details zur Registrierung finden Sie auf den entsprechenden [Google-Seiten](https://www.google.com/recaptcha) (<https://www.google.com/recaptcha>). Wenn Sie das Schlüsselpaar registriert haben, tragen Sie dieses in die `mycore.properties` wie folgt ein:

```
#####
### Google - ReCaptcha (https://www.google.com/
recaptcha) ## registered for: www.mycore.de ###
#####
MIR.ReCaptcha.secret.key=1234hierkommtdannderprivateschlueselhin4321
MIR.ReCaptcha.site.key=5678undhierderoeffentlicheschluesel8765
```

Diese Konfiguration ist so auch auf [mycore.org/mir/](http://www.mycore.org/mir/) (<http://www.mycore.org/mir/>) aktiv und kann dort getestet werden.



## 4 Aktualisierung

---

Für die Aktualisierung werden im Normalfall nur das `mir.war` und ggf. die solr-Konfigurationsdateien `solrconfig.xml` und `schema.xml` ausgetauscht und danach die Webanwendung inkl. Solr neu gestartet. Sollten Konfigurationen angepasst oder Daten migriert werden müssen, werden entsprechende Informationen an dieser Stelle bereitgestellt.