

MyCoRe Installation mit IBM CM8.3

- Kurzanleitung -

Kathleen Krebs
Ulrike Krönert
Jens Kupferschmidt
Holger König

4. November 2005

Inhaltsverzeichnis

1. Voraussetzungen für eine MyCoRe Anwendung.....	3
2. Download und Installation des MyCoRe Kerns.....	3
3. Die MyCoRe Beispielanwendung.....	4
4. Installation der Webanwendung.....	8

Vorwort

Dieses Dokument beinhaltet eine Kurzanleitung zur Installation von MyCoRe mit IBM CM8.3. Ein bereits installiertes CM-System 8.3, DB2 8.2, WebSphere 5.1.1.3, Java 1.4 und NSE 8.2 mit den jeweils aktuellen Fixpacks wird vorausgesetzt. Grundlage für die hier beschriebene Installation ist die MyCoRe Version 1.2 vom 28. September 2005 .

1. Voraussetzungen für eine MyCoRe Anwendung

Es wird davon ausgegangen, dass ein MyCoRe-Benutzer angelegt ist. Dieser wird im Folgenden als **mcradmin** bezeichnet.

Zum Betrieb von MyCoRe sind zuerst einmal die folgenden freien Komponenten nötig:

1. Java SDK 1.4 oder höher (in Kombination mit CM8.3 und WebSphere wird empfohlen das WebSphere-Java zu verwenden)
2. bash (erforderlich unter Unix-Systemen für die Unixtools der Beispielanwendung)
3. cvs-Client (andernfalls muss MyCoRe unter www.mycore.de heruntergeladen werden)
4. apache-ant 1.6.x

Ausserdem sollte hinsichtlich der im Rahmen des Content Managers installierten Komponenten nachfolgende Punkte geprüft werden, da sich diese in der Vergangenheit häufig als Ursache vieler Probleme während der Installation zeigten.

- Verwendet der Benutzer **mcradmin** die richtige Javaversion?
- Wurde die DB2 korrekt installiert und sind alle Patches eingespielt?
- Sind NSE und DB2 gestartet und ein Zugriff auf die DB2 von der User-ID **mcradmin** aus erlaubt (siehe dazu auch Abschnitt 2)?
- Ist der Content Manager inklusive NSE richtig installiert und sind alle Patches eingespielt?
- Hat der Content Manager Verbindung zur NSE?
- Ist WebSphere richtig installiert und sind alle Patches drin?
- Sind alle Konfigurationen für Apache und WebSphere gemacht?
- Ist der Ressource Manager richtig konfiguriert und läuft die entsprechende WebSphere Applikation?

2. Download und Installation des MyCoRe Kerns

Es wird empfohlen einen eigenen MyCoRe-Benutzer anzulegen, z.B. mcradmin. Dieser Nutzer muss Datenbankkommandos ausführen können. Dazu ist folgende Zeile in die .profile-Datei einzufügen (wobei der Benutzer db2inst1 hier exemplarisch für den db2-Standardbenutzer steht):

```
. ~db2inst1/sqllib/db2profile
```

Nun sollte es möglich sein mit dem MyCoRe-Benutzer eine Verbindung zur DB2 herzustellen, konkret benötigt MyCoRe die Verbindung zur Datenbank des Libraryserver (standardmäßig ist dies die ICMNLSDB). Der Aufruf des Befehls:

```
> db2 connect to icmnlsdb
```

Sollte daher erfolgreich sein (weitere Hinweise siehe Punkt 3.7)

Die folgenden Schritte sind mit dem MyCoRe-Benutzer an gewünschter Stelle im Dateisystem (z.B. das Home-Verzeichnis von mcradmin /home/mcradmin) durchzuführen.

1. Download der aktuellen Version unter www.mycore.de oder via cvs

```
> cvs -d :pserver:anoncvs@server.mycore.de:/cvs checkout -r  
release_1_2_fixes mycore
```

Die .tar.gz version von www.mycore.de muß mit GNUtar ausgepackt werden, weil sonst aufgrund der langen Pfadenamen verstümmelt werden

2. Setzen der Variable MYCORE_HOME als Pfad zum MyCoRe-Verzeichnis.

```
( MYCORE_HOME = /home/mcradmin/mycore )
```

3. Für die Installation muss ausserdem ANT_HOME und JAVA_HOME (vorzugsweise auf das WebSphere-Java) gesetzt sein.

4. Anpassen der Konfigurationsdatei für die Installation,

```
$MYCORE_HOME/config/build.properties, wie folgt:
```

- Auskommentieren der JDBCStore-Einträge für DB2 Version 7* und 8* (ggf. die Pfade anpassen, anstatt db2jcc.jar kann in älteren Installationen auch noch runtime.zip vorhanden sein und verwendet werden)

```
# IBM DB2 Version 8.2 under IBM AIX without Hibernate
# Use package: org.mycore.backend.sql
# Hint: the JDBC driver is in the DB2 package
MCR.JDBCStore.Type=db2
MCR.JDBCStore.BaseDir=/home/db2inst1/sqllib
MCR.JDBCStore.Jars=java/sqlj.zip java/db2java.zip java/db2jcc.jar
MCR.JDBCStore.ClassesDirs=function
MCR.JDBCStore.LibPath=java lib
```

- Einkommentieren des JDBCStore-Eintrages für Hibernate

```
# Hibernate for Linux or Windows
# Use package: org.mycore.backend.hibernate
# Hint: the Hibernate access to SQL DB's is included!
# MCR.JDBCStore.Type=hibernate
```

- Auskommentieren der XMLStore-Einträge für Content Manager Version 8 (ggf. die Pfade anpassen)

```
# IBM Content Manager Version 8 under IBM AIX
MCR.XMLStore.Type=cm8
MCR.XMLStore.BaseDir=/opt/IBMcm8
MCR.XMLStore.Jars=lib/cm8*.jar lib/dtappsrv.jar
MCR.XMLStore.ClassesDirs=lib cmgmt
MCR.XMLStore.LibPath=lib
```

- Einkommentieren der XMLStore-Einträge für JDOM

```
# Native JDOM store in memory
# MCR.XMLStore.Type=jdom
```

5. Aufruf von 'ant info' in \$MYCORE_HOME, um die aktuelle Konfiguration zu überprüfen.

6. Kompilieren und packen des MyCoRe-Kerns mit dem Aufruf von 'ant jar'. Dies erzeugt in

```
$MYCORE_HOME/lib die Datei mycore-for-cm8.jar
```

Anmerkung: Weitere mögliche Befehle können mit 'ant usage' angezeigt werden.

3. Die MyCoRe Beispielanwendung

Die Beispielanwendung wird in das gleiche Verzeichnis wie der MyCoRe-Kern (im Beispiel dlwww) gespielt.

1. Download des DocPortals unter www.mycore.de oder via cvs

```
> cvs -d :pserver:anoncvs@server.mycore.de:/cvs checkout -r
release_1_2_fixes docportal
```

2. Setzen der Umgebungsvariablen \$DOCPORTAL_HOME

```
( DOCPORTAL_HOME = /home/mcradmin/docportal )
```

3. DocPortal enthält anwendungsinstanzübergreifende Konfigurationsdateien, wie das Grundlayout der Webseiten, Datentypen etc. Fürs erste reicht die Anpassung der `mycore.properties.private` (siehe Punkt 4.) und der `ContentStoreSelectionRules.xml` im `$DOCPORTAL_HOME/config`-Verzeichnis:

- Wenn Dokumente standardmäßig im ResourceManager gespeichert werden sollen, dann muß der Defaultwert auf CM8 gesetzt werden:

```
<!-- Default file content store, if no other rule applies: -->
<ContentStoreSelectionRules
    default="CM8"
>
```

- Einzelne Dateitypen können ebenfalls dem CM8 oder anderen Systemen zugewiesen werden (internal Filesystem FS, IBM Video Charger VC usw.)

```
<store ID="CM8">
  <type> html </type>
  <type> msppt </type>
  <type> msword95 </type>
  <type> msword97 </type>
  <type> pdf </type>
  <type> ps </type>
  <type> rtf </type>
  <type> txt </type>
  <type> xml </type>
  <type> css </type>
  <type> ppt </type>
  <type> jpg </type>
  <type> gif </type>
  <type> jpeg </type>
  <type> png </type>
  <type> tiff </type>
</store>
```

- Den Lucene Store auskommentieren.

Anmerkung: Der Content Manager sollte die angegebenen MimeTypes unterstützen. CSS beispielsweise wird nicht standardmäßig unterstützt. Der MimeType muss angelegt werden (mittels AdminClient) wenn Dateien dieses Types vom CM verwaltet werden sollen.

4. Die eigentliche Konfiguration der Beispielanwendung geschieht in der Datei `mycore.properties.private` im `$DOCPORTAL_HOME/config`-Verzeichnis:

- Kopieren von `mycore.properties.private.template` nach `mycore.properties.private`

- Anpassen der Datei:

1. wenn WebSphere als Application Server verwendet wird, auskommentieren der Zeile `#MCR.session.param=;jsessionid=0000`
2. alle Einträge im Abschnitt `# The persistence configuration for CM8` auskommentieren
3. das Passwort von icmadmin setzen
`MCR.persistence_cm8_password=?????????`
4. Prüfen der Pfade für die Volltextindizierung:
`MCR.persistence_cm8_textsearch_indexdir=`
`/home/db2inst1/sqlllib/db2ext/indexes`
`MCR.persistence_cm8_textsearch_workingdir=`
`/home/db2inst1/sqlllib/db2ext/indexes`

5. bei der Konfiguration des Resource-Managers gibt es eine alternative Konfiguration, hier entweder memory oder none auswählen (empfohlen ist memory)

```
##MCR.IFS.ContentStore.CM8.StoreTemp.Type=none
#MCR.IFS.ContentStore.CM8.StoreTemp.Type=memory
```

6. JDBC-Treiber für DB2 angeben, HSQLDB-Treiber einkommentieren:

```
# JDBC parameters for connecting to DB2
MCR.persistence_sql_database_url=jdbc:db2:ICMNLSDb
MCR.persistence_sql_driver=COM.ibm.db2.jdbc.app.DB2Driver

# JDBC parameters for connecting to HSQLDB
#MCR.persistence_sql_database_url=jdbc:hsqldb:hsqldb://localhost:9001
#MCR.persistence_sql_driver=org.hsqldb.jdbcDriver
#MCR.persistence_sql_database_userid=sa
#MCR.persistence_sql_database_passwd=
```

7. Die SQL-Klassendefinitionen von Hibernate auf DB2 umstellen

```
# Hibernate SQL classes definition
# [...] alles einkommentieren
```

```
# Native SQL classes definition
[...] alles auskommentieren
```

8. Setzen der Host-Adresse für den Remotezugriff von anderen MyCoRe-Instanzen:

```
MCR.remoteaccess_selfremote_host=4wayb.boeblingen.de.ibm.com
```

9. Anpassen der Pfade und eMail-Adresse im Workflow-Bereich. Damit das Versenden von eMails funktioniert, muss ausserdem ein gültiger smtp-Server angegeben werden:

```
# Generic mail configuration for MCRMailer
MCR.mail.server=4wayb.boeblingen.de.ibm.com
MCR.mail.protocol=smtp
MCR.mail.debug=false
```

10. Für das Java-Applet müssen Schlüssel generiert werden, die Konfiguration dazu geschieht mittels den "Applet JAR signing Parameters". Wichtig ist das Anpassen des Speicherortes der Schlüssel:

```
# File where the keys are stored in
SIGN.KeyStore=/home/mcradmin/docportal/keystore/keys
```

11. Abschließend muss noch die Konfiguration für den Indexbrowser angepasst werden. Da es bei der hier angegebenen Konfiguration keine MCRXMLTABLE geben wird, muss eine andere Tabelle angegeben werden:

```
# Objects for MCRObjectID (document)
MCR.IndexBrowser.robots.Table=MCRXMLDOC
[...]

# Objects for MCRObjectID (author)
MCR.IndexBrowser.robota.Table=MCRXMLAUTH
```

5. Auch hier können mit dem Aufruf von 'ant info' in \$DOCPORTAL_HOME die Konfiguration nochmals geprüft werden.

6. Als nächstes folgt das Erzeugen der XML-Schemadateien, das Kompilieren zusätzlicher Klassen für die Beispielanwendung und das Generieren der Kommandozeilenskripte:

```
ant create.schema
ant jar
ant create.scripts
```

7. Danach können die Datenstrukturen angelegt werden:

- Test des db2-Zugriffs für den Benutzer, der MyCoRe installiert:

```
> db2 connect to icmnlbdb
```

Ist der Aufruf nicht erfolgreich, muss mcradmin entsprechende Rechte bekommen bzw. die Konfiguration des Benutzers angepasst werden. Dazu muss mit dem DB2-Benutzer (meist

db2inst1) folgendes Kommando ausgeführt werden:

```
> db2 grant dbadm on database to user mcradmin
```

Für die nächsten Schritte muss dieser Test erfolgreich sein. Danach kann die Verbindung terminiert werden:

```
> db2 connect reset
```

- Anlegen der Tabellen und CM-ItemTypes mit 'ant create.metastores'. Dies kann eine Weile dauern (je nach System bis zu einer Stunde)!

Anmerkung: Hier scheint es noch Probleme mit den Skripten beim Anlegen des Index zu geben. Es kommt dann zu einem DB2-Fehler und das Skript wird abgebrochen. Ein erneuter Aufruf von 'ant create.metastores' ist notwendig. Dies kann mehrfach der Fall sein.

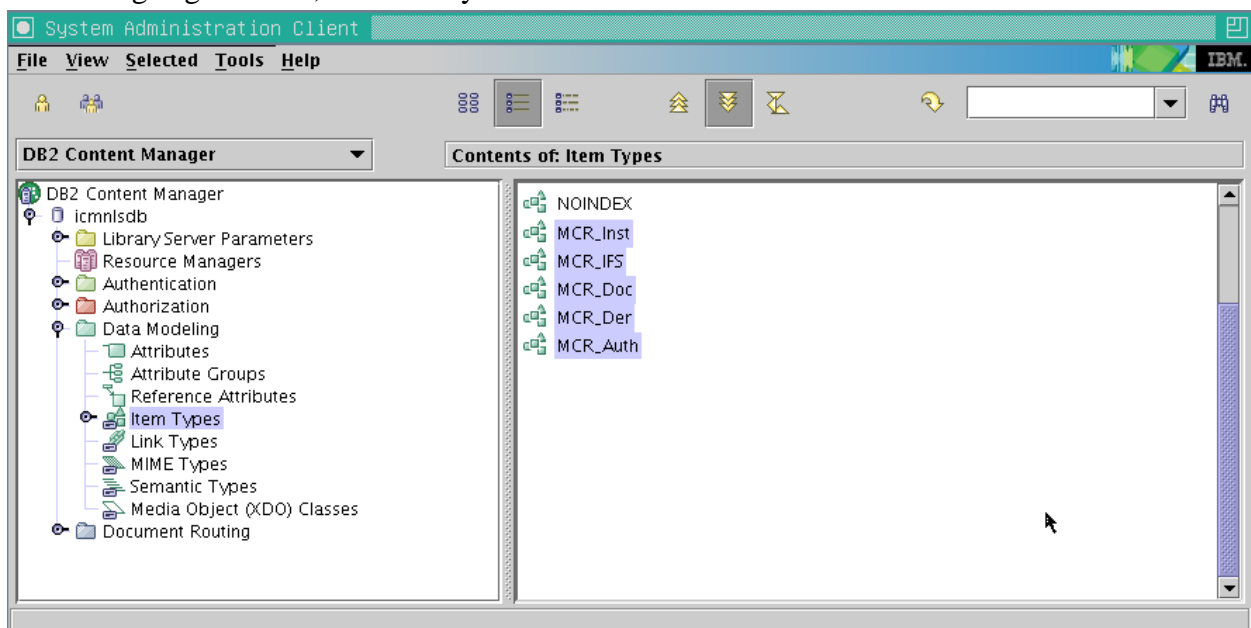
- Prüfen ob alle Tabellen angelegt worden

```
> db2 connect to icmnlbdb
```

```
> db2 list tables
```

Table/View	Schema	Type	Creation time
MCRCateg	MCRADMIN	T	2005-09-19
MCRCategLab	MCRADMIN	T	2005-09-19
MCRClass	MCRADMIN	T	2005-09-19
MCRClassLab	MCRADMIN	T	2005-09-19
MCRcStore	MCRADMIN	T	2005-09-19
MCRFSNodes	MCRADMIN	T	2005-09-19
MCRGROUPADMINS	MCRADMIN	T	2005-09-19
MCRGROUPMEMBERS	MCRADMIN	T	2005-09-19
MCRGROUPS	MCRADMIN	T	2005-09-19
MCRID	MCRADMIN	T	2005-09-19
MCRLinkClass	MCRADMIN	T	2005-09-19
MCRLinkHref	MCRADMIN	T	2005-09-19
MCRPRIVS	MCRADMIN	T	2005-09-19
MCRPRIVSLOOKUP	MCRADMIN	T	2005-09-19
MCRUSERS	MCRADMIN	T	2005-09-19
MCRXMLAUTH	MCRADMIN	T	2005-09-19
MCRXMLDER	MCRADMIN	T	2005-09-19
MCRXMLDOC	MCRADMIN	T	2005-09-19
MCRXMLINST	MCRADMIN	T	2005-09-19

- Prüfen ob alle ItemTypes (MCR_Inst, MCR_IFS, MCR_Doc, MCR_Der und MCR_Auth) angelegt worden, mit dem Systemadministrationsclient:



8. Abschliessend werden noch Benutzer und Klassifikationen der Beispielanwendung geladen:
 - Anlegen der MyCoRe-Benutzer mit dem Aufruf 'ant create.users'
 - Anlegen der Klassifikationen mit dem Aufruf 'ant create.class'
9. In `$DOCPORTAL_HOME/modules/module-wcms/aif/config` die Datei `mycore.properties.wcms.template` nach `mycore.properties.wcms` kopieren und die Pfade anpassen.
10. Bevor die Webanwendung erstellt werden kann, muss ein Schlüsselpaar für das JavaApplet generiert werden, mittels 'ant create.genkeys'
11. 'ant war' erstellt eine in WebSphere installierbare Webanwendung, die auf den soeben erstellten Datenbestand zugreift.

4. Installation der Webanwendung

Für die Installation der Webanwendung muss die Administrationskonsole gestartet sein:

```
> /usr/WebSphere/AppServer/bin/startServer.sh server1
```

Läuft die Administrationskonsole, kann im Webbrowser folgende URL für die weitere Installation aufgerufen werden: `http://<hostname>:9090/admin`

Nach der Anmeldung sollte als erstes ein neuer Anwendungsserver¹ erstellt werden, da so eine eigenständige Konfiguration der Umgebungsvariablen möglich ist:

1. **Servers > Application Servers**
2. **New**



3. Den Anweisungen folgen. Normalerweise können die Defaulteinstellungen belassen werden. Als Name kann beispielsweise **docportal** verwendet werden.

Danach folgt die eigentliche Installation der Webanwendung² `docportal.war`:

1. **Applications > Install New Application**
2. als **Server Path** ist der Pfad zum vorher erstellten Datei `docportal.war` einzutragen
3. **Context Root** muss angegeben werden, z.B. `docportal`
4. **next** klicken
5. Schritt 1: **next** klicken
4. Schritt 2: **next** klicken

¹ http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.base.doc/info/aes/ae/trun_svr_create.html

² http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.base.doc/info/aes/ae/trun_app_inst_wiz.html

5. Schritt 3: der Webanwendung den vorher angelegten Anwendungsserver zuweisen, in diesem Beispiel: **docportal** in der Checkbox anwählen, dann die Zeile mit **server=docportal** und anwenden **apply**
6. **next** klicken
7. Schritt 4: Angaben prüfen und Installation mit **finish** beenden

Nun folgt die Konfiguration des Anwendungsservers:

1. **Servers > Application Servers**
2. den gewünschten Server auswählen, hier **docportal**
3. im unter **Additional Properties** die **Process Definition > Process Execution** wählen

Weitere Merkmale	
Java Virtual Machine	Erweiterte Einstellungen für die JVM.
Prozessausführung	Merkmale, die die Ausführung des Betriebssystemprozesses steuern, wie z. B. RunAs-Berechtigungen, Umask, Prozesspriorität.
Prozessprotokolle	Merkmale, die die Weiterleitung der nativen Ein-/Ausgabedatenströme von Prozessen steuern.
Umgebungseinträge	Eine Liste mit Einträgen, die der Ausführungsumgebung für den Prozess hinzugefügt werden soll.
Überwachungs-Policy	Merkmale, die die Überwachung des Prozesses durch den Node Agent steuern.
Protokollierung und Tracing	Protokoll- und Trace-Einstellungen für diesen Server festlegen.

4. als Benutzer den MyCoRe-Benutzer angeben, hier **mcradmin**
5. Gruppe auf die entsprechend zugehörige Gruppe, z.B. **users** setzen
6. Änderungen mit **ok** übernehmen
8. es folgt die Konfiguration der **Java Virtual Machine**
9. als **classpath** die folgenden Pfade einfügen (ggf. anpassen)


```
/home/db2inst1/sqlllib/java12/db2java.zip:
/usr/lpp/cmb/lib/cmbsdk81.jar:
/usr/lpp/cmb/cmgmt
```

Anmerkung: Ich habe hier den CLASSPATH der beim Aufruf von 'ant info' im \$DOCPORTAL_HOME ausgegeben wird eingetragen, da einige Klassen sonst nicht gefunden werden.

10. Änderungen mit **ok** übernehmen

Damit die Sitzungsverwaltung³ von MyCoRe uneingeschränkt und auch ohne Cookies funktioniert, muss nun noch das Umschreiben von URLs aktiviert werden:

1. Dazu muss nochmals der entsprechende Anwendungsserver ausgewählt werden **Servers > Application Servers** und hier **docportal**
2. im Abschnitt **Additional Properties** den Link **Web Container** und dann **Session Management** anklicken
3. in diesem Fenster **Enable URL Rewriting** und **Enable protocol switch rewriting** aktivieren
4. auch diese Änderung muss bestätigt werden durch klicken auf **ok**

³ http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/topic/com.ibm.websphere.base.doc/info/aes/ae/tpsr_sesa.html



Abschließend müssen alle Änderungen gespeichert und das Plugin für den Webserver erstellt werden:

1. Es wird bereits darauf hingewiesen, dass es Änderungen gab und diese gespeichert werden müssen. In diesem Text einfach auf **save** klicken
2. Nun das Speichern bestätigen mit **save**
3. Das Plugin kann unter **Environment > Update Web Server Plugin** aktualisiert werden, Bestätigung mittels **ok**
11. Beenden der Sitzung mit **Logout**

Als nächstes muss der Application Server gestartet werden:

```
> /usr/WebSphere/AppServer/bin/startServer.sh docportal
```

Beim ersten Mal startet der Server nicht, da einige Logdateien noch nicht geschrieben werden können. Dies liegt daran, dass die Webanwendung als **mcradmin** ausgeführt wird. Daher sind nun folgende Schritte nötig:

1. > `chown -R mcradmin:mcr /usr/WebSphere/AppServer/logs/ docportal`
2. > `mkdir /usr/WebSphere/AppServer/tranlog/ docportal`
3. > `chown -R mcradmin:mcr /usr/WebSphere/AppServer/tranlog/ docportal`
4. > `chown -R mcradmin:mcr /usr/WebSphere/AppServer/installedApps/<host>/docportal.ear`
5. > `chmod 666 /usr/WebSphere/AppServer/logs/activity.log`

Danach ist der Server noch einmal zu stoppen und neu zu starten:

```
> /usr/WebSphere/AppServer/bin/stopServer.sh docportal
> /usr/WebSphere/AppServer/bin/startServer.sh docportal
```

Jetzt sollten Sie auf das DocPortal unter der URL `http://<hostname>/docportal` zugreifen können.