

# Die Speicherung der MyCoRe-Metadaten-Objekte

2016.06 2017.06

by Kathleen Neumann, Jens Kupferschmidt  
2018-01-03

## Table of contents

1 Der native MyCoRe-Metadaten-Speicher.....	2
2 Speicherung mit Versionierung.....	2



## 1 Der native MyCoRe-Metadaten-Speicher

MyCoRe unterscheidet zwischen dem primären Datenspeicher für alle Metadaten und die daraus ableitbaren sekundären Metadaten wie z. B. die Link-Beziehungen (MCRLinkTableManager), der SOLR-Index usw. Die primären Objekt-Metadaten werden in nativer XML-Form direkt in einer verzeichnisstruktur auf der Festplatte gespeichert. Dies hat den Vorteil, dass auch eine Datensicherung mit File Level backup erfolgen kann, ohne auf spezielle Datenbank-Backups zurückgreifen zu müssen. Alle sekundären Daten können mit Repair-Kommandos wieder erstellt werden (siehe dazu *Laufender Betrieb -> Backup und Recovery*). Die Implementierung dieses Speichers wird im MyCoRe als IFS2 (Internal File System 2) bezeichnet.

Soll nur der native Speicher genutzt werden, so ist die folgende Store-Klasse zu konfigurieren.

```
MCR.Metadata.Store.DefaultClass=org.mycore.datamodel.ifs2.MCRMetadataStore
```

Jede Projekt-ID und jeder Objekttyp werden in einem eigenen IFS2 Metadata Store angelegt. In der Konfiguration muss dazu nur das Basisverzeichnis angegeben werden.

```
MCR.Metadata.Store.BaseDir=%MCR.datadir%/metadata
```

Für jedes Projekt und Objekttyp entsteht ein separates Unterverzeichnis.

Die Standardkonfiguration speichert dann z. B. die Metadaten des Objektes

*MyMIR\_mir\_07910403* in der Datei

```
%MCR.datadir%/metadata/MyMIR/mir/0791/04/
```

```
MyMIR_mir_07910403.xml
```

IFS2 Stores bilden Unterverzeichnisse (Slots), um die Dateien gleichmässiger im Filesystem zu verteilen. Diese Unterverzeichnisse werden aus der ID abgeleitet, anhand des vorkonfigurierten Slot Layouts. Das Standard Slot Layout ist **n-2-2** wobei n die Anzahl der Stellen in der MyCoRe Object ID - 4 ist. Bei `MCR.Metadata.ObjectID.NumberPattern=00000000` ist das Standard Layout 4-2-2, achtstellige Objekt-IDs (4+2+2=8), die ersten vier Ziffern der ID bilden die erste Verzeichnisebene, die nächsten zwei Ziffern der ID bilden die zweite Verzeichnisebene unterhalb des Ordners für Projekt-ID und Objekttyp.

## 2 Speicherung mit Versionierung

Neben der nativen Speicherung ist es auch möglich eine Versionierung der Metadaten durchzuführen. Diese läuft parallel zum Prozess der nativen Speicherung, benötigt aber etwas mehr Zeit als die native Variante alleine. D. h. mit dieser Implementierung lassen sich Änderungen an Metadaten-Objekten verfolgen, alte und gelöschte Versionen der Metadaten wiederherstellen. Diese Implementierung ist bei Leseoperationen auf die aktuellste Version der Metadaten genau so schnell, bei create/update/delete Operationen aber deutlich langsamer. Um die Versionierung einzuschalten sind die folgenden Properties erforderlich.

Soll die SVN-Versinierung mit genutzt werden, so ist die folgende Store-Klasse zu konfigurieren.

```
MCR.Metadata.Store.DefaultClass=org.mycore.datamodel.ifs2.MCRVersioningMetadataStore
```

Der Speicherort der Versionsdaten ist wie folgend anzugeben:

```
MCR.Metadata.Store.SVNBase=file:///MCR.datadir%/versions-metadata
```

Diese SVN Repositories können wie andere auch mit externen Subversion Kommandos befragt werden, aber es ist keine lokale Subversion Installation erforderlich. Die benötigte Funktionalität wird von MyCoRe mitgeliefert.

Es ist möglich, die Einstellungen zum verwendeten Store, Basisverzeichnis und Slot Layout individuell für einzelne Objekttypen und Projekt-IDs zu überschreiben.

So könnte z. B. für einen bestimmten Objekttyp (hier MyMIR\_mir) Versionierung aktiviert werden, für andere Objekttypen aber der schnellere MCRMetadataStore weiter genutzt werden.

```
MCR.Metadata.Store.DefaultClass=org.mycore.datamodel.ifs2.MCRMetadataStore
```

```
MCR.IFS2.Store.MyMIR_mir.Class=org.mycore.datamodel.ifs2.MCRVersioningMetadataStore
```

```
MCR.IFS2.Store.MyMIR_mir.SlotLayout=4-2-2
```

```
MCR.IFS2.Store.MyMIR_mir.BaseDir=/path/to/metadata/store/mir/
```

```
MCR.IFS2.Store.MyMIR_mir.SVNRepositoryURL=file:///path/to/local/svn/repository/  
mir/
```

Wenn MCRVersioningMetadataStore verwendet wird, wird das Metadatenobjekt zunächst in einer XML-Datei gespeichert, die dann im Anschluss via SVN commit versioniert wird. Standardmäßig wird der Zeitstempel der Datei nachträglich auf den exakten Zeitpunkt des SVN commits gesetzt. Dies kann unter Linux zu Fehlern aufgrund von Zugriffsrechten führen. In diesem Fall kann die Zeitstempel-Synchronisation abgeschaltet werden durch:

```
MCR.IFS2.SyncLastModifiedOnSVNCommit=false
```