



---

# Developer Guide

---

RELEASE 1.3

15. November 2006

**Frank Lützenkirchen (Essen/Duisburg)**

**Jens Kupferschmidt (Leipzig)**

**Andreas Trappe, Thomas Scheffler (Jena)**

**Kathleen Krebs (Hamburg)**

**Anja Schaar (Rostock)**

## Vorwort

Dieses Dokument ist für die aktiven und interessierten Entwickler gedacht. Es dokumentiert alle im Architecture Board getroffenen Vereinbarungen. Die hier beschriebenen Festlegungen sind für alle am Entwicklungsprozess Beteiligten verbindlich. Sie sollen vor allem die einheitliche Gestaltung des Projektes garantieren.

# Inhaltsverzeichnis

1	Festlegungen für die Code Entwicklung.....	1
1.1	Vorbemerkungen.....	1
1.2	Encoding.....	1
1.2.1	Allgemeines.....	1
1.2.2	Konfiguration unter Eclipse.....	1
1.3	Code Formatierung.....	1
1.3.1	Allgemeines.....	1
1.3.2	Konfiguration unter Eclipse.....	1
1.4	Kommentare im Code.....	2
1.4.1	Allgemeines.....	2
1.4.2	Kommentare im Java-Code.....	2
1.4.3	Kommentare in den Stylesheets.....	4
1.4.4	Logging.....	4
2	Dokumentation.....	5
2.1	Allgemeines.....	5
3	Anhang.....	6
3.1	Abbildungsverzeichnis.....	6
3.2	Tabellenverzeichnis.....	7



# 1 Festlegungen für die Code Entwicklung

## 1.1 Vorbemerkungen

Von der Entwicklergruppe wird das Werkzeug Eclipse zur Arbeit am MyCoRe-Projekt empfohlen. Es enthält sowohl Funktionalitäten zur Integration von CVS wie auch zur Qualitätssicherung des zu erstellenden JAVA-Codes.

## 1.2 Encoding

### 1.2.1 Allgemeines

Grundsätzlich geht MyCoRe davon aus, dass alle Dateien, die nicht sprachabhängig sind, mit **UTF-8** kodiert sind. Dies gestattet eine gute Nutzung auch über die Grenzen des deutschsprachigen Raumes hinaus. Dies betrifft vor allem die Dateitypen:

- Java-Code - \*.java
- XML-/XSL-Dateien - \*.xml

### 1.2.2 Konfiguration unter Eclipse

Die Einstellung erfolgt in Eclipse im jeweiligen Projekt unter:

1. Properties --> Info --> Text file encoding --> UTF-8

## 1.3 Code Formatierung

### 1.3.1 Allgemeines

Um bei der Arbeit mit dem CVS-System nur inhaltliche Änderungen zu erfassen und diese nicht mit Umformatierungen zu verwechseln, wird der gesamte Code einheitlich nach folgenden Regeln erstellt:

- Formatierung gemäß den Java-Konventionen
- Tabulatoren werden durch Leerzeichen ersetzt
- Einrückung mit vier Zeichen
- Zeilenumbruch und maximale Zeilenlänge 160

### 1.3.2 Konfiguration unter Eclipse

Die Einstellung erfolgt in Eclipse im jeweiligen Projekt unter:

1. Properties --> Java code style --> Formatter

2. dort Java Conventions als Vorlage wählen
3. Indentation --> Tab Policy --> Spaces only
4. Indentation --> Indentation size --> 4
5. Indentation --> Tab size --> 4
6. Line wrapping --> Maximum line width --> 160
7. Die Vorlage wird dann als mycore gespeichert.

## **1.4 Kommentare im Code**

### **1.4.1 Allgemeines**

Alle Kommentare sind in Englisch abzufassen. Dabei ist auf allgemein verständliche Sprachkonstrukte zu achten. Der Kommentar soll die codierten Vorgänge gut beschreiben und für andere nachvollziehbar machen.

### **1.4.2 Kommentare im Java-Code**

Jede Java-Quelltext-Datei hat das nachfolgende Aussehen. Das gestattet ein einheitliches Auftreten des Projektes. Da für alle Dateien mittels JavaDoc automatisch eine Dokumentation generiert wird, ist es Pflicht, die erstellte Klasse mit einer allgemeinen Beschreibung zum Zweck und Einsatz der Klasse zu versehen. Weiterhin sind alle public oder protected Methoden mit einer Beschreibung zu versehen. Hierzu gehört auch die Dokumentation der übergebenen Parameter, der Rückgabewerte und ggf. der geworfenen Exceptions.

```
/*
 * $RCSfile: MCR....java,v $
 * $Revision: 1.49 $ $Date: 2006/05/17 11:49:32 $
 *
 * This file is part of *** M y C o R e ***
 * See http://www.mycore.de/ for details.
 *
 * This program is free software; you can use it, redistribute it
 * and / or modify it under the terms of the GNU General Public License
 * (GPL) as published by the Free Software Foundation; either version 2
 * of the License or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program, in a file called gpl.txt or license.txt.
 * If not, write to the Free Software Foundation Inc.,
 * 59 Temple Place - Suite 330, Boston, MA 02111-1307 USA
 */

package org.mycore.datamodel.metadata;

import java.io.File;

/**
 * This class implements all methode for ...
 *
 * @author Jens Kupferschmidt
 * @version $Revision: 1.49 $ $Date: 2006/05/17 11:49:32 $
 */
final public class MCR... {

    /**
     * This method ..
     *
     * @param a The paremeter a is the first value.
     * @return a if it is a positive number, else return 0.
     */
    public final int abs(a) {
    }
}
```

Abbildung 1.1: Mindestkommentar im Java Quellcode

### 1.4.3 Kommentare in den Stylesheets

Auch alle XSLT-Dateien sollten kurze Kommentare enthalten. Unbedingt erforderlich sind auf jeden Fall die Zeilen für die Versionskontrolle. Angestrebt wird folgendes Aussehen eines Stylesheets.

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

<!-- ===== -->

<!-- $Revision: 1.2 $ $Date: 2006/10/12 11:52:14 $ -->

<!-- ===== -->

<xsl:stylesheet version="1.0"

    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

>

...

</xsl:stylesheet>
```

Abbildung 1.2: Kommentare für XSLT Dateien

### 1.4.4 Logging

Alle Logging-Informationen werden, sofern nicht eine Umsetzung mittels der Internationalisierung I18N erfolgt, in Englisch notiert. Für MyCoRe ist das **log4j**-Paket des Apache-Projektes zu verwenden. Es gibt 4 definierte Log-Level mit nachfolgenden Bestimmungen. Es ist davon auszugehen, dass eine normale Anwendung allgemein auf den Level INFO gesetzt ist.

- ERROR – Gibt Informationen zu nicht behebbaren Fehlern, z.B. Exceptions, zurück.
- WARN – Gibt Informationen zu Fehlern zurück, welche die Weiterarbeit der Anwendung nicht ausschließen. In der Regel wird mit Standardwerten weitergearbeitet.
- INFO – Gibt allgemeine Informationen für den normalen Anwender bzw. die Log-Datei aus. Diese Nachrichten haben nur informativen Charakter.
- DEBUG – Gibt zusätzliche Informationen, die gezielt durch Einschalten dieses Levels abgerufen werden sollen, aus.



## 2 Dokumentation

### 2.1 Allgemeines

Dokumentationen zum MyCoRe-Projekt sind als Open Office Dokument zu erstellen und in PDF-Form zu speichern. Die Sprache der Dokumente ist wahlweise Deutsch oder Englisch.

## 3 Anhang

### 3.1 Abbildungsverzeichnis

Abbildung 1.1: Mindestkommentar im Java Quellcode.....	3
Abbildung 1.2: Kommentare für XSLT Dateien.....	4

## 3.2 Tabellenverzeichnis

Tabelle 2.1: Softwarevoraussetzungen.....	5
Tabelle 2.2: Umgebungsvariablen unter Windows.....	6
Tabelle 2.3: Umgebungsvariablen unter Unix.....	7
Tabelle 3.1: Aufbau des MyCoRe-Kerns.....	13
Tabelle 4.1: Beispieldaten im CVS.....	19
Tabelle 5.1: Beispielgruppen in Docportal.....	20
Tabelle 5.2: Beispielbenutzer in Docportal.....	21
Tabelle 5.3: Parameter des MCRStartEditorServlets.....	30
Tabelle 5.4: Mögliche Aktionen mit dem MCRStartEditorServlet auf dem Plattenbereich.	31
Tabelle 5.5: Mögliche Aktionen mit dem MCRStartEditorServlet im Server.....	32
Tabelle 6.1: Feste Test-Instanzen für das MyCoRe-Beispiel.....	38
Tabelle 7.1: Aufbau der MCRObjektID.....	47
Tabelle 7.2: MyCoRe-Basisdatentypen.....	52