

Code Style Guide

Formatting Rules

The code style conventions used in the project are based on the [style-guide defined](#) of Fedora-3. They should prevent commits cluttered with format changes.

Here are the major rules:

1. Four space indents for Java, and 2-space indents for XML. NO TABS
2. K&R style braces

```
if (code) {  
    // code  
} else {  
    // code  
}
```

3. Do not use wildcard imports
4. Write Javadocs for public methods and classes. Keep it short and to the point
5. Avoid public instance variables; use accessors
6. Use public methods sparingly; implementation details are not public
7. Maximum length of lines is 120 characters.
8. Create Javadocs for types of *at least* the following descriptivity

```
/**  
 * @author Joe Developer  
 * @since MMM DD, YYYY  
 */  
public class MyClass
```

9. Each source file should contain a license header much like the following:

```
/**  
 * Copyright 2015 DuraSpace, Inc.  
 *  
 * Licensed under the Apache License, Version 2.0 (the "License");  
 * you may not use this file except in compliance with the License.  
 * You may obtain a copy of the License at  
 *  
 *     http://www.apache.org/licenses/LICENSE-2.0  
 *  
 * Unless required by applicable law or agreed to in writing, software  
 * distributed under the License is distributed on an "AS IS" BASIS,  
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 * See the License for the specific language governing permissions and  
 * limitations under the License.  
 */
```

Use the [maven-license-plugin](#) to check for and add missing headers:

Check for missing headers

```
mvn license:check
```

Add missing headers

```
mvn license:format
```

IDE Setups

IDE settings are located in the project source. IDE users are strongly recommended to apply these formatting settings.

- Eclipse settings are here: [fcrepo4/src/site/eclipse](#)
- IntelliJ settings TBD (purportedly, [this plugin](#) will let you use the Eclipse settings above within IntelliJ)

Checkstyle

We're in the process of adding [checkstyle](#) enforcement to our modules (meaning, if you violate some of the major style rules, the build will fail).

To check for violations, run the following command:

```
mvn checkstyle:check
```