

Whitelabel Group

DITA Tool Chain

Reference Document

Imprint

Whitelabel Reference Document

Version: 0.7.0
Release Version: 0.1.0
Release Date: 19th April, 2017

Document Language: en
Document Status: preview
Security Level: Public

Last Change: April 2017

Copyright 2017, Whitelabel Group

Whitelabel Group

Whiting Street 101, Sampletown

Legal Notice

All company logos are a registered trademark of Whitelabel Group.

The product names mentioned in this documentation are either trademarks or registered trademarks of the respective owners and are stated for identification purposes only.

This documentation and the software components are protected by copyright 2017, Whitelabel Group.

Contents

Imprint.....	ii
Legal Notice.....	iii
1 Introduction.....	5
1.1 Context.....	5
1.2 Scope.....	5
1.3 Target Audience.....	5
2 Reference.....	6
2.1 Tables.....	6
2.2 Notes.....	6
2.3 References.....	7
2.4 Figures.....	7
2.5 Lists.....	9
2.6 Sections.....	9
2.7 Samples.....	9
3 Reference Bill of Material.....	11
3.1 Whitelabel Reference Document 0.7.0.....	11
3.2 Bill of Materials.....	12
3.3 Licenses in Effect.....	15
3.4 Copyright Notices.....	15
4 Part Introduction.....	17
4.1 Part Content.....	17
A Appendix.....	18
A.1 Maven Support.....	18
A.2 DITA Conventions.....	18
List of Tables.....	19
List of Figures.....	20
Glossary.....	21
Index.....	22

1 Introduction

Getting started with *Darwin Information Typing Architecture (DITA)* and the *DITA Open Toolkit (DITA-OT)* is quite a challenge. Advantages over other documentation approaches can only be achieved once a complete infrastructure - a tool chain - is defined, established, and constantly managed.

This document is part of the {metæffekt} DITA Tool Chain. It demonstrates how content can be organized and transformed to different outputs, such as *Hypertext Markup Language, version 5 (HTML5)* and *Portable Document Format (PDF)*.

The content of this document is used as reference to demonstrate the features and capabilities of the tool chain. As such the content is often artificial.

This short introduction already illustrates the power of the toolchain to a great extent. Mind the references to the glossary. The glossary is context specific to the document that is produced and based on existing term definitions.

Glossary terms use key references. This means a reference that does not point directly to an existing file, but an identifier (key) that is then mapped to a file from the ditamap.

1.1 Context

The {metæffekt} DITA Tool Chain envisions to lower the hurdle for an organization to access a consistent and conceptually sound configuration that can be adapted to organization-specific requirements more easily.

Once the fundamentals are established the tool chain can be integrated deeper in the procedure and processes of the organization. By generating DITA content from other sources such as a issue tracker, development resources (such as Maven POMs or other project definitions) generated content can be integrated into the documents.

The ultimate goal of the {metæffekt} DITA Tool Chain is to automate the complete release documentation of a software development project. This supports incremental software development approaches.

1.2 Scope

The document scope is two-fold. On a general level the motivation for the {metæffekt} DITA Tool Chain is conveyed.

Second, and more prominent, features of the tool chain are explained and highlighted. The document is regarded as a reference and example for what can be done with tool chain.

1.3 Target Audience

This document is intended for technical writers and developers at the same time. It is regarded as essential that in particular these two groups better collaborate on writing technical documentation about the software project in focus.

2 Reference

The document serves as reference for selected features of DITA and the metaeffekt DITA Tool Chain. Primarily it shows how DITA content can be organized. As the result can be rendered to a PDF document and HTML5 the reference also serves as test and demonstration of the resulting document styling.

2.1 Tables

There are several ways to create a table using *DITA-OT*. In general the simpletable is quite easy to use:

Column 1	Column 2
Monday	Get up early.
Tuesday	Get up late.
Wednesday	Stay in bed.

However, the CALS Table is much more flexible and gives you more options.

Column 1	Column 2
Thursday	Visit Uncle Joe
Friday	Get back on track.
Saturday	Gain faith.

Table 1: CALS Table Title

Please note that the CALS table has a title. This title can be included in the table references in the backmatter of the document. Check it out.

In case you are using an WYSIWYG¹ editor there is no reason to not use the CALS table.

Please also mind that both tables are rendered differently. Actually the {metaeffekt} DITA-OT customization does currently not pay particular attention to using simple table.

2.2 Notes

Notes enable to highlight information.



Note: This is a note without further specified attributes.

Using a paragraph in a note is also possible and gives a nice result.

Yet another paragraph.

The importance attribute is currently not evaluated by the toolkit. However, the type of the note influences the graphics and the label.

¹ WYSIWYG: What You See Is What You Get.



Attention: This is note with type attention.

You should define a clear semantics of what the different types mean in your organization.



CAUTION: This is note with type caution.

You should define a clear semantics of what the different types mean in your organization.



DANGER: This is note with type danger.

You should define a clear semantics of what the different types mean in your organization.



Fastpath: This is note with type fastpath.

You should define a clear semantics of what the different types mean in your organization.



Important: This is note with type attention.

You should define a clear semantics of what the different types mean in your organization.

2.3 References

When writing DITA content it is possible to use all kinds of references. For example it is possible to point to a link in the web using a href. For example <http://metaeffekt.org>. You can also point to another chapter. For example [Introduction](#) on page 5.

Really useful are key-based references (keyref). Basically you use an alias for the reference. This alias must be mapped within the ditamap. For example [Introduction](#) on page 5. This can be explicitly useful when content is made reusable. The reusable piece want's to introduce a reference. But the reference is dangling until the content is reused in a ditamap. Then the reference must of course exist. The ditamap has to specify how the link is to be resolved.

2.4 Figures

This chapter demonstrates the usage of figures.

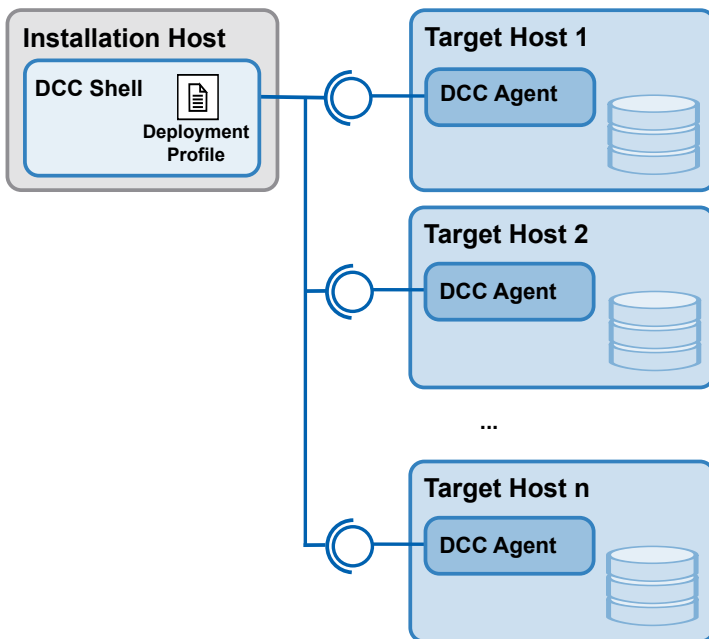


Figure 1: Using an SVG as figure.

SVGs - as these are XML files - can use placeholders as well. This way context specific replacements can be performed. In addition, this approach represents an option combine a custom layout with content without the need to write XSLT.

SVGs can be scaled as required. Giving best results with any print output.



Figure 2: Using a PNG as figure.

Non-scalable image formats have a defined resolution. On screen the resolution may still be sufficient (72dpi), but when zooming into or when printing low resolution pictures show artifacts.



Figure 3: Using an JPG as figure.

2.5 Lists

1. Item 1
 2. Item 2
- Item 1
 - Item 2

2.6 Sections

Title One

First content.

Title Two

Second content.

2.7 Samples

This chapter shows how to embed snippets and code samples.



Figure 4: Showing the output of a command line tool.

```
<pluginManagement>
  <plugins>
    <plugin>
      <groupId>org.metaeffekt.dita</groupId>
      <artifactId>ae-dita-maven-plugin</artifactId>
      <version>0.7.0</version>
    </plugin>
  </plugins>
```

```
</pluginManagement>
```

Figure 5: Showing some XML content.

Unfortunately no syntax-highlighting is intergrated yet.

```
</**
 * A {@link Binding} is capable of connecting two capabilities (usually a provided with a required
 * {@link Capability}) with each other. Please note that the compatibility of the {@link Capability}
 * (shared {@link CapabilityDefinition}) is not enforced. The implementation however expects that
 * the attribute keys of the target are covered by the source {@link Capability}.
 *
 * @author Karsten Klein
 */
public class Binding extends AbstractElement {
    ...
}
```

Figure 6: Showing some JAVA content.

3 Reference Bill of Material

The following chapters illustrates how content of a Bill of Material is rendered. The content was extracted from another metaeffekt project. The content is not relevant for the reference project, but is only used here for illustration purposes.

3.1 Whitelabel Reference Document 0.7.0

The *software distribution* covered by this document is based on third-party *components* restricted by individual terms and conditions defined in the respective licenses and/or contracts.

This document is part of the *extended software distribution* which addresses the obligations deduced from the effective license terms and conditions. The software distribution and extended software distribution must always be made available, distributed, or transferred in combination.

3.1.1 Commercial Components

The covered software distribution contains commercial third-party components. Commercial third-party software components are generally licensed under vendor-specific license terms and specific contractual agreements with the licensee.

License and contract information regarding commercial licenses are covered in section . Standard license texts are included in the extended software distribution when permitted by the terms and conditions of the license and/or associated contract.

3.1.2 Open Source Components

In general, for each and every open source component the individual terms and conditions of the associated license apply. The licensee of an open source component is granted a non-exclusive right of use by the respective right holders restricted by the stipulated terms and conditions defined in the associated license. In particular, liability and warranty regulations of the individual open source license apply only in relation to the respective right holder.

The license files and notice files of the *effective license* are provided as extended scope of the software distribution.

License obligations of the individual open source components, as far as these can be addressed in documentation, are detailed in section . In case a license demands the provision of source code for a component, the source code is made available by adding the source code artifacts to the extended software distribution or by making the source available as individually outlined in the notices.

3.1.3 Import / Export Information

This distribution does not include any components containing cryptographic software. No known restrictions apply.

3.1.4 Import/Export Information

This distribution includes components containing cryptographic software. Please note, that the country the software is consumed in may have restrictions on the import, possession, and/or use of cryptographic software. Furthermore, the export or re-export of this distribution to other countries may be restricted or even prohibited, due to the included cryptographic components.

Before using or re-exporting this distribution check the country specific laws, regulations and policies. For more information, see also <http://www.wassenaar.org>.

The following table provides details on the cryptographic software included in this distribution:

Component Name	Import/Export Classification Details
Bouncy Castle	The Bouncy Castle artifacts contain cryptographic software and include additional crypto providers. Classification details can be found online: http://www.bouncycastle.org/wiki/display/JA1/Frequently+Asked+Questions .
Jasypt	Java Simplified Encryption API (Jasypt) uses the Java Cryptography Extension API and as such is classified for export control. See details online: http://www.jasypt.org/faq.html#what-is-jasypts-export-classification .
Oracle JRE	The Oracle Java Runtime Environment (Oracle JRE) contains cryptographic software. For more details consult the Oracle Binary Code License agreement on http://www.oracle.com/technetwork/java/javase/terms/license/index.html . For further details on the Oracle Global Trade Compliance check http://www.oracle.com/us/products/export/index.html .

Table 2: Cryptographic Components

Please note that `$(cryptographic.information.provider)` does not take any responsibility for the correctness, actuality, and content of the above external links. The links were last verified on `$(cryptographic.link.verification.date)`.

3.2 Bill of Materials

Subsequently all included third-party components and individual *artifacts* are listed with references to the licenses and/or contracts.

3.2.1 Third-Party Artifacts

Group Id	Artifact Id	Version	License
	bootstrap-theme (css)	3.3.2	MIT License
	bootstrap (css)	3.3.2	MIT License
	bootstrap (js)	3.3.2	MIT License
	glyphicons-halflings (eot)	3.3.2	MIT License
	glyphicons-halflings (svg)	3.3.2	MIT License
	glyphicons-halflings (ttf)	3.3.2	MIT License
	glyphicons-halflings (woff)	3.3.2	MIT License
	glyphicons-halflings (woff2)	3.3.2	MIT License
	jquery (js)	1.11.1	MIT License
ant-contrib	ant-contrib	1.0b3	Apache License Version 1.1
cglib	cglib	3.2.4	Apache License Version 2.0
com.google.guava	guava	19.0	Apache License Version 2.0
commons-codec	commons-codec	1.10	Apache License Version 2.0
commons-collections	commons-collections	3.2.2	Apache License Version 2.0
commons-fileupload	commons-fileupload	1.3.2	Apache License Version 2.0
commons-io	commons-io	2.5	Apache License Version 2.0
commons-lang	commons-lang	2.6	Apache License Version 2.0
jline	jline	2.12.1	BSD 3-Clause License
net.jcip	jcip-annotations	1.0	Creative Commons Public Domain
org.apache.ant	ant-commons-logging	1.9.7	Apache License Version 2.0 (with-subcomponents)

Group Id	Artifact Id	Version	License
org.apache.ant	ant-launcher	1.9.7	Apache License Version 2.0 (with-subcomponents)
org.apache.ant	ant	1.9.7	Apache License Version 2.0 (with-subcomponents)
org.apache.camel	camel-core	2.16.2	Apache License Version 2.0
org.apache.camel	camel-restlet	2.16.2	Apache License Version 2.0
org.apache.camel	camel-spring	2.16.2	Apache License Version 2.0
org.apache.commons	commons-exec	1.3	Apache License Version 2.0
org.apache.commons	commons-lang3	3.4	Apache License Version 2.0
org.apache.httpcomponents	httpclient	4.5.2	Apache License Version 2.0
org.apache.httpcomponents	httpmime	4.5.2	Apache License Version 2.0
org.apache.httpcomponents	httpcore	4.4.5	Apache License Version 2.0
org.apache.james	apache-mime4j-core	0.7.2	Apache License Version 2.0
org.apache.logging.log4j	log4j-api	2.7	Apache License Version 2.0
org.apache.logging.log4j	log4j-core	2.7	Apache License Version 2.0
org.apache.logging.log4j	log4j-slf4j-impl	2.7	Apache License Version 2.0
org.apache.velocity	velocity	1.7	Apache License Version 2.0
org.bouncycastle	bcpkix-jdk15on	1.55	Bouncy Castle License
org.bouncycastle	bcprov-jdk15on	1.55	Bouncy Castle License
org.ow2.asm	asm	5.0.4	BSD 3-Clause License
org.restlet.jee	org.restlet.ext.fileupload	2.3.1	Apache License Version 2.0
org.restlet.jee	org.restlet.ext.httpclient	2.3.1	Apache License Version 2.0
org.restlet.jee	org.restlet.ext.servlet	2.3.1	Apache License Version 2.0
org.restlet.jee	org.restlet	2.3.1	Apache License Version 2.0
org.slf4j	jcl-over-slf4j	1.7.21	MIT License
org.slf4j	jul-to-slf4j	1.7.21	MIT License
org.slf4j	slf4j-api	1.7.21	MIT License
org.springframework.shell	spring-shell	1.1.0. RELEASE	Apache License Version 2.0
org.springframework	spring-aop	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-beans	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-context-support	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-context	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-core	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-expression	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)
org.springframework	spring-tx	4.3.2. RELEASE	Apache License Version 2.0 (with-subcomponents)

3.2.2 Third-Party Components

Component	File(s)	License
Apache Ant Contrib	ant-contrib-1.0b3.jar	Apache License Version 1.1
Apache Ant	ant-commons-logging-1.9.7.jar ant-launcher-1.9.7.jar ant-1.9.7.jar	Apache License Version 2.0 (with-subcomponents)
Apache Camel	camel-core-2.16.2.jar camel-restlet-2.16.2.jar camel-spring-2.16.2.jar	Apache License Version 2.0
Apache Commons Codec	commons-codec-1.10.jar	Apache License Version 2.0
Apache Commons Collections	commons-collections-3.2.2.jar	Apache License Version 2.0
Apache Commons Exec	commons-exec-1.3.jar	Apache License Version 2.0
Apache Commons File Upload	commons-fileupload-1.3.2.jar	Apache License Version 2.0
Apache Commons IO	commons-io-2.5.jar	Apache License Version 2.0
Apache Commons Lang	commons-lang-2.6.jar commons-lang3-3.4.jar	Apache License Version 2.0
Apache HTTP Components	httpclient-4.5.2.jar httpcore-4.4.5.jar httpmime-4.5.2.jar	Apache License Version 2.0
Apache James Mime4J	apache-mime4j-core-0.7.2.jar	Apache License Version 2.0
Apache Log4J	log4j-api-2.7.jar log4j-core-2.7.jar log4j-slf4j-impl-2.7.jar	Apache License Version 2.0
Apache Velocity	velocity-1.7.jar	Apache License Version 2.0
ASM	asm-5.0.4.jar	BSD 3-Clause License
Bootstrap	bootstrap-theme-3.3.2.min.css bootstrap-3.3.2.min.css bootstrap-3.3.2.min.js glyphicons-halflings-regular.eot glyphicons-halflings-regular.svg glyphicons-halflings-regular.ttf glyphicons-halflings-regular.woff glyphicons-halflings-regular.woff2	MIT License
Bouncy Castle	bcpkix-jdk15on-1.55.jar bcprov-jdk15on-1.55.jar	Bouncy Castle License
GCLib	cglib-3.2.4.jar	Apache License Version 2.0
Google Guava	guava-19.0.jar	Apache License Version 2.0
JCip Annotations	jcip-annotations-1.0.jar	Creative Commons Public Domain
JLine	jline-2.12.1.jar	BSD 3-Clause License
JQuery	jquery-1.11.1.min.js	MIT License
Restlet	org.restlet.ext.fileupload-2.3.1.jar org.restlet.ext.httpclient-2.3.1.jar org.restlet.ext.servlet-2.3.1.jar org.restlet-2.3.1.jar	Apache License Version 2.0

Component	File(s)	License
SLF4J	jcl-over-slf4j-1.7.21.jar jul-to-slf4j-1.7.21.jar slf4j-api-1.7.21.jar	MIT License
Spring Framework	spring-aop-4.3.2.RELEASE.jar spring-beans-4.3.2.RELEASE.jar spring-context-support-4.3.2.RELEASE.jar spring-context-4.3.2.RELEASE.jar spring-core-4.3.2.RELEASE.jar spring-expression-4.3.2.RELEASE.jar spring-tx-4.3.2.RELEASE.jar	Apache License Version 2.0 (with-subcomponents)
Spring Shell	spring-shell-1.1.0.RELEASE.jar	Apache License Version 2.0

3.3 Licenses in Effect

License
Apache License Version 1.1
Apache License Version 2.0
Apache License Version 2.0 (with-subcomponents)
BSD 3-Clause License
Bouncy Castle License
Creative Commons Public Domain
MIT License

3.4 Copyright Notices

Apache License Version 1.1

Component	Group Id / Artifact Id	Version	Notice
Apache Ant Contrib	ant-contrib / ant-contrib	1.0b3	This product includes software developed by the Ant-Contrib project (http://sourceforge.net/projects/ant-contrib).

Apache License Version 2.0 (with-subcomponents)

Component	Group Id / Artifact Id	Version	Notice
Apache Ant	org.apache.ant / ant	1.9.7	This distribution includes Apache Ant. Apache Ant is provided under the terms and conditions of the Apache License Version 2.0 with further ammendments covering licenses of sub-components contained in the Apache Ant binaries. Apache Ant uses artifacts provided under the W3C Software Notice and License; SAX2 (public domain). Please check the details in the license and notice files provided with this distribution.

Component	Group Id / Artifact Id	Version	Notice
Spring Framework	org.springframework / spring-core	4.3.2.RELEASE	This distribution includes the Spring Framework. The Spring Framework is provided under the terms and conditions of the Apache License Version 2 with further amendments of licenses of sub-components contained in the distributed package. Spring Framework uses ASM 4.0 (BSD 3-Clause License) Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved; CGLIB 3.0 (Apache License Version 2).

BSD 3-Clause License

Component	Group Id / Artifact Id	Version	Notice
JLine	jline / jline	2.12.1	This distribution includes JLine which is provided under the terms and conditions of the BSD 3-Clause License. Copyright (c) 2002-2016, the original author or authors. All rights reserved.
ASM	org.ow2.asm / asm	5.0.4	This distribution includes ASM which is provided under the terms and conditions of the BSD 3-Clause License. Copyright (c) 2000-2011 INRIA, France Telecom. All rights reserved.

Bouncy Castle License

Component	Group Id / Artifact Id	Version	Notice
Bouncy Castle	org.bouncycastle / bcpkix-jdk15on	1.55	This distribution includes Bouncy Castle provided under the Bouncy Castle License. Copyright (c) 2000-2016 The Legion of the Bouncy Castle Inc. (http://www.bouncycastle.org).

MIT License

Component	Group Id / Artifact Id	Version	Notice
Bootstrap	bootstrap	3.3.2	This distribution includes Bootstrap provided under the MIT License. Copyright (c) 2011-2016 Twitter, Inc.
Bootstrap	glyphicons-halflings	3.3.2	This distribution includes Bootstrap provided under the MIT License. Copyright (c) 2011-2016 Twitter, Inc.
JQuery	jquery	1.11.1	This distribution includes JQuery provided under the MIT License. Copyright (c) 2005, 2014 jQuery Foundation, Inc.
SLF4J	org.slf4j / jcl-over-slf4j	1.7.21	This distribution includes SLF4J. SLF4J is provided under the terms and conditions of the MIT License. SLF4J is Copyright (c) 2004-2014 QOS.ch. All rights reserved.

Part



4 Part Introduction

This is the introduction of the part.

4.1 Part Content

This is a the content of a part. Usually there would be much more content, such that it is adequate to organize the document is parts.

A Appendix

A.1 Maven Support

Apache Maven is the supported way to use the {metæffekt} DITA Toolchain. No further software prerequisites are required. A customized DITA-OT Toolkit is used implicitly when using the metaeffekt DITA Plugin.

A.1.1 Installing Apache Maven

Apache Maven is the primary prerequisite for using the {metæffekt} DITA Tool Chain.

1. Download the latest Apache Maven 3.3 release from <http://maven.apache.org/download.cgi>.



Note: JAVA Required.

JAVA 1.7 is required to run Apache Maven 3.3. In general Oracle JAVA 1.8 is required by the {metæffekt} DITA Tool Chain.

In case JAVA is not available, it can be downloaded from <https://java.com/en/download/>.

2. Follow the instruction provided on <http://maven.apache.org/install.html>

Test the installation was successful by executing the command

```
mvn -version
```

on the console. When Apache Maven was successfully installed the command provides the following output:

```
Apache Maven 3.3.9 ...
```

A.1.2 Maven Configuration

Maven uses XML project model descriptors also known as POM files. The structure of these POM files is defined on LINK.

To enable the metaeffekt DITA Plugin specify the following plugin:

The configuration expects a ditamap at the location `src/main/dita/${project.artifactId}`

A.2 DITA Conventions

The {metæffekt} DITA Tool Chain uses numerous conventions to structure content within a documentation project.

List of Tables

Table 1: CALS Table Title.....	6
Table 2: Cryptographic Components.....	12

List of Figures

Figure 1: Using an SVG as figure.....	8
Figure 2: Using an PNG as figure.....	8
Figure 3: Using an JPG as figure.....	9
Figure 4: Showing the output of a command line tool.....	9
Figure 5: Showing some XML content.....	9
Figure 6: Showing some JAVA content.....	10

Glossary

PDF

Portable Document Format developed by Adobe®.

Artifact

In general an artifact is a physical file. An artifact can be an executable binary, an archive, a document or a simple configuration files.

Component

A component is a group of one or more artifacts. While artifacts are physical files the component can be regarded as logical aggregate. Licensing information applies on component-level.

Darwin Information Typing Architecture (DITA)

OASIS defined standard promoting the creation of standard information types and domain specific markup. DITA defined content can be aggregated in a DITA map and rendered to different output formats such as PDF, HTML or RTF.

DITA Open Toolkit

The DITA-OT supports to render DITA-based content to different output formats. The DITA-OT is an open source endeavor that supports all available versions of the DITA specification.

Effective Licenses

There are various cases where a component is made available by a provider under multiple licenses. The effective licenses are selected licenses relevant for redistribution.

- A component is provided with a dual or multiple licensing option where either one or the other license can be used for redistribution. In this case a single effective license needs to be determined.
- A component is provided with a license in a specific or later version. In this case a version of the license has to be selected as effective license.
- For a component, which contains subcomponents the license of the component and the individual licenses of the subcomponents are effective.

Extended Software Distribution

An extended software distribution contains all required licensing information and notices to fulfill the obligations when distributing the associated software distribution. An extended distribution lists covered third-party components and - if stipulated by the license terms and conditions - the license texts, notices and source code artifacts.

HTML5

Hypertext Markup Language, version 5

Software Distribution

A software distribution is the resulting archive that is packaged for delivering the software to a recipient. Depending on context and relationship between the provider and recipient of a distribution the license terms and conditions of the covered components have to be addressed.

Index

P

Power [5](#)