# Advanced features of OpenCms 6

Alkacon Software GmbH

March, 2006

# Index

- Alkacon OpenCms 6 technical overview
- Creating OpenCms templates with taglibs
- Using and configuring the integrated full text search engine
- Managing multilingual websites
- Defining structured content using XML schema
- Integrating "Direct Edit" functionality

# OpenCms 6 technical overview

- **OpenCms has a strict 3-tier architecture**
  - Workplace / Website front-end
  - OpenCms Core "middleware" logic
  - Database access layer separated in "drivers"
- **Shell access to the OpenCms database if no web GUI is available**
- **Module API for pluggable extensions**
- **Content in is stored in XML structures**
- **Flexible, file based data repository (VFS)**

# OpenCms 6 technical overview (2)

- OpenCms leverages the power of many proven OSS components
  - Lucene (search engine)
  - Dom4J (XML API)
  - Jtidy (HTML parser)
  - Doomdark JUG (UID Generator)
  - Log4j (Log file handler)
  - Digester (XML Configuration reader)
  - Commons DBCP/Pool (DB Connection)
  - Quartz (Scheduler)
  - Junit (Unit testing)
  - …

# Creating OpenCms templates with taglibs

- **OpenCms Templates are simple JSP pages**
  - Taglib for accessing OpenCms functions
  - Can be combined with other taglibs (e.g. JSTL)
  - All taglib functions are also available in scriptlet API
- **Templates are simple resources in the OpenCms VFS, this means they**
  - are version controlled
  - are workflow enabled (online / offline)
  - can be distributed in modules

# Creating OpenCms templates with taglibs (2)

- Templates can have multiple sub-elements, for example:
  - Navigation element
  - Ad banner element
  - Article list element
- A template usually formats / displays the content of one xmlpage/xmlcontent
  - However, content can also be used on multiple locations
- The same templates can also be used to format the output of other JSP pages

# Creating OpenCms templates with taglibs (3)

**Simple template example:**

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms"%>

<html>
<head>
<title>
OpenCms - <cms:property name="Title" escapeHtml="true" />
</title>
</head>
<body>

<cms:include element="body" />

</body>
</html>
```

# Demo Screenshot: A sample template

# Using and configuring the full text search engine

- **OpenCms integrates the powerful Lucene full text search engine**

- **Creation of multiple indices is supported**
  - Example: One index per language

- **Included are extractors for various commonly used document types**
  - E.g. Word, Excel, PDF, XML, TXT, Html…

- **All searched content is also permission controlled, i.e. users will only see the results where they are allowed to "read"**

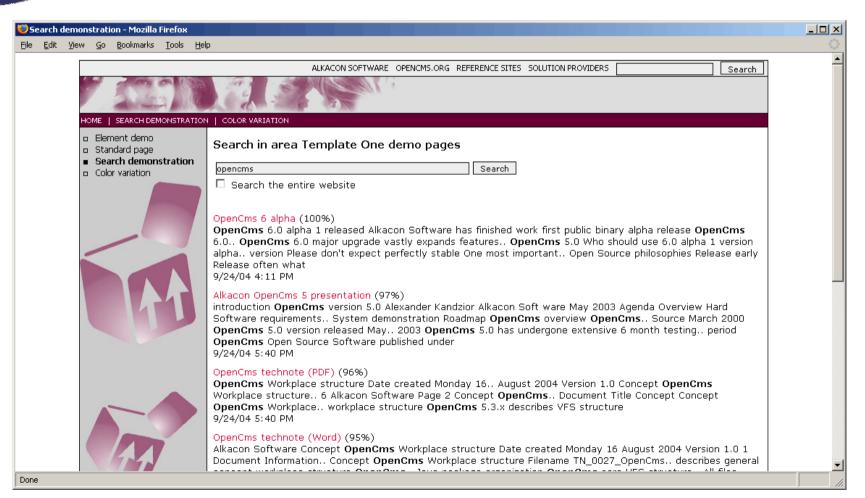# Using and configuring the full text search engine (2)

- Configuration is done in the file **`opencms-search.xml`**
- Full control over the output layout, default layout shows a Google-like ranking with highlighted search terms
  - Highlight function can also be configured
- Also searches in properties (meta information)
- Restricted searches to subtrees of the VFS possible
  - E.g. for "search only in this department"

# Demo Screenshot:
# Full text search output

# Managing multilingual websites

- OpenCms 6 editable content is mainly composed of two types
  - xmlpage (unstructured, WYSIWYG content)
  - xmlcontent (structured with XML schema)
- Both types have a common XML structure to allow creation of content in multiple languages
- The selected language is controlled by the configured *locale handler*
  - default: configure locale property on folder
  - alternative: use browser locale

# Managing multilingual websites (2)

**Sample XML Page structure:**

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

<pages
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
      "http://www.opencms.org/dtd/6.0/xmlpage.xsd">
 <page language="de">
   <element name="text1">
    <links/>
    <content><![CDATA[<H1>Dieser Text ist in Deutsch</H1>]]></content>
   </element>
 </page>
 <page language="en">
   <element name="text1">
    <links/>
    <content><![CDATA[<H1>This text is in English</H1>]]></content>
   </element>
 </page>
</pages>
```

# Managing multilingual websites (3)

## Sample XML Content structure:

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

<Articles
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation=
      "opencms://sites/default/xmlcontent/article.xsd">

  <Article language="en">
   <Title><![CDATA[Sample Article 1]]></Title>
   <Teaser><![CDATA[This is the first sample article.]]></Teaser>
   <Text name="element0">
    <links/>
    <content><![CDATA[<p>This is the article 1 text</p>content>
   </Text>
  </Article>
</Articles>
```

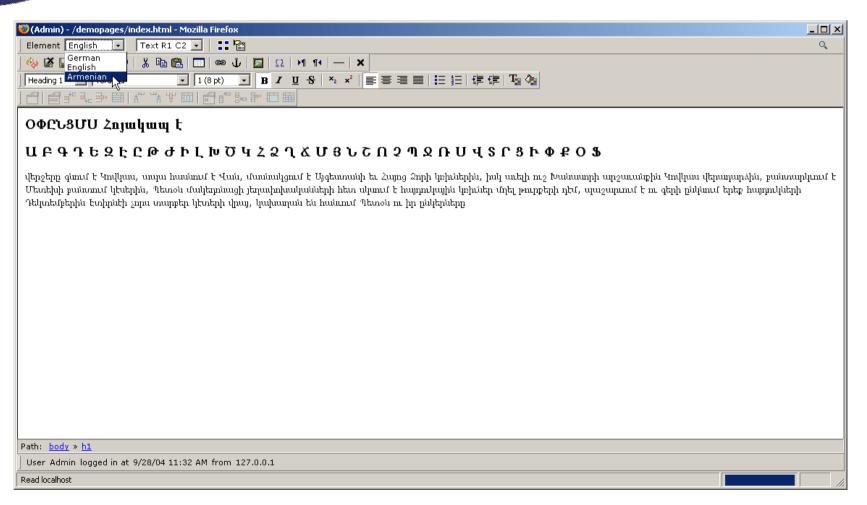# Managing multilingual websites (4)

- Content in OpenCms can be created in any charset encoding supported by Java
  - UTF-8 is the default in version 6
- Different encodings can be mixed in one installation
  - One page could be using ISO-8859-1 while the others on the site use UTF-8
- The WYSIWYG editor supports automatic conversion of "unknown" chars to HTML entities (and vice-versa)
  - E.g. &#8364; for "€" symbol in ISO-8859-1

# Demo Screenshot:
# Multilingual editing in UTF-8

# Defining structured content using XML schema

- XML content is one of the most powerful features of OpenCms 6
- This allows simple creation of "Structured content" according to a w3c XML schema
  - The XML schema used must itself follow another XML schema for xmlcontents
- The XML schema used is also managed in the OpenCms VFS
- An editor for the xmlcontent is automatically created using the schema

# Defining structured content using XML schema (2)

**Sample XML Content schema:**

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

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

    <xsd:include schemaLocation="opencms://opencms-xmlcontent.xsd"/>

    <xsd:element name="Articles" type="OpenCmsArticles"/>
    <xsd:complexType name="OpenCmsArticles">
            <xsd:sequence>
            <xsd:element name="Article" type="OpenCmsArticle" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="OpenCmsArticle">
            <xsd:sequence>
                            <xsd:element name="Title" type="OpenCmsString" />
                            <xsd:element name="Teaser" type="OpenCmsString" />
                            <xsd:element name="Text" type="OpenCmsHtml" />
            </xsd:sequence>
            <xsd:attribute name="language" type="OpenCmsLocale" use="required"/>
    </xsd:complexType>
    <xsd:annotation>
            <xsd:appinfo>
                            <mapping element="Title" mapto="property:Title" />
            </xsd:appinfo>
    </xsd:annotation>
</xsd:schema>
```

# Defining structured content using XML schema (3)

## Sample XML Content structure:

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

<Articles
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation=
        "opencms://sites/default/xmlcontent/article.xsd">

  <Article language="en">
    <Title><![CDATA[Sample Article 1]]></Title>
    <Teaser><![CDATA[This is the first sample article.]]></Teaser>
    <Text name="element0">
      <links/>
      <content><![CDATA[<p>This is the article 1 text</p>content>
    </Text>
  </Article>
</Articles>
```

# Defining structured content using XML schema (4)

- XML content can also be mapped to corresponding VFS properties or attributes
  - E.g. "Title" field to "Title" property
- Some XML content types are predefined in OpenCms
  - Work currently in progress
  - Predefined content types can be extended through an open API / configuration
- A taglib API is available to access XML content in lists, detail pages etc.
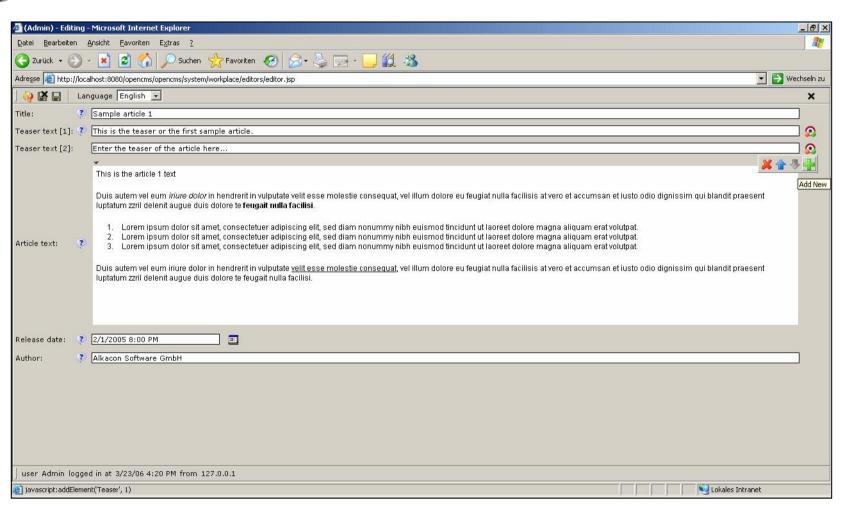
# Defining structured content using XML schema (5)

**Sample XML Content list page:**

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>
<cms:include property="template" element="head" />

<cms:contentload collector="allInFolder"
    param="/article_${number}.html|article">

<div class="element2">
<b><cms:contentshow element="Title" /></b><br>
<cms:contentshow element="Teaser" /><br>
<small><a href="<cms:link><cms:contentshow
    element="${opencms.filename}" /></cms:link>">read
    more...</a></small>
</div>

</cms:contentload>

<cms:include property="template" element="head" />
```

# Demo Screenshot:
# Structured content in XML Schema

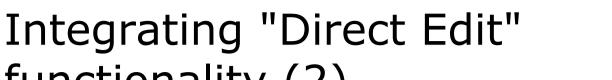# Integrating "Direct Edit" functionality

- "Direct edit" allows for access to the editor through the website "front end"
- All access is permission/lock controlled
  - No access if no "write" permission
  - Access disabled if file is currently locked
- "Direct edit" is available only in the Offline project, not on the published Online site
- Integration is very simple with just some additional tags
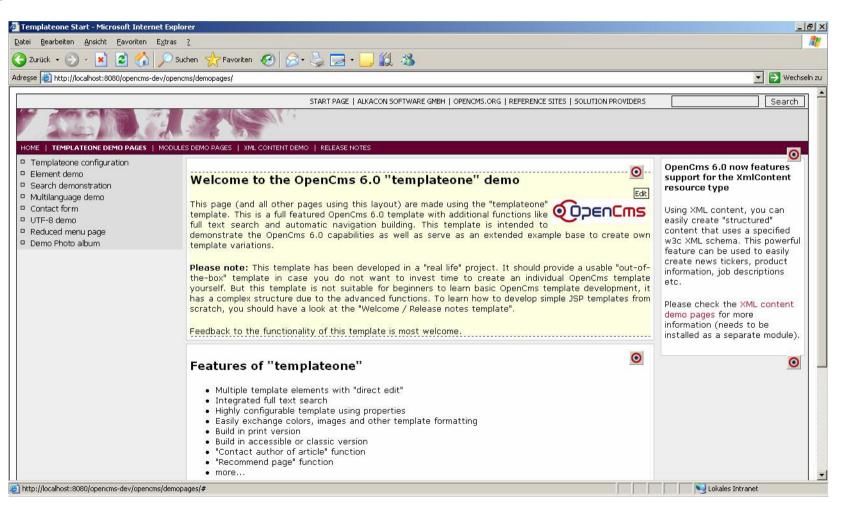  - HTML inserted can optionally be selected

# Integrating "Direct Edit" functionality (2)

## "Direct edit" for a template example:

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms"%>

<html>
<head>
<title>
OpenCms - <cms:property name="Title" escapeHtml="true" />
</title>
<cms:editable />
</head>
<body>

<cms:include element="body" editable="true" />

</body>
</html>
```

# Demo Screenshot:
# Direct edit from the website

# Integrating "Direct Edit" functionality (3)

**"Direct edit" on a XML content example:**

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms"%>

…

<cms:contentload collector="singleFile" param="${opencms.uri}"
    editable="true">

<h1><cms:contentshow element="Title" /></h1>
<p>
<cms:contentshow element="Text" />
</p>

</cms:contentload>

…
```

# Alkacon OpenCms Contacts

Alkacon Software GmbH

An der Wachsfabrik 13

50996 Cologne

Germany

http://www.alkacon.com

http://www.opencms.org