



## OpenCms Days 2009

Technical Track:

***Using the Alkacon OAMP Module series  
to enhance OpenCms***

Michael Emmerich,  
Alkacon Software GmbH.



1. Introduction of the Alkacon OAMP Modules
  - Overview over exiting OAMP Modules
  
2. OAMP DocCenter Module
  - Usage of the DocCenter Module
  - Excursion : The ResourceInit Handlers
  
3. OAMP Webform Module and its extensions
  - Webform Module
  - Survey Module
  - Comments Module



# Introduction of the Alkacon OAMP Modules



## What are the Alkacon OAMP Modules?

- OAMP = OpenCms Add-On Modul Paket
- Free, Open Source Modules to enhance OpenCms
- Source Code available via CVS
- Partly based on experience of customer projects
- New versions of several Modules will be released after the OpenCms days



# Overview over existing Alkacon OAMP Modules



# Alkacon OAMP Newsletter Module



## Key Features:

- Configurable mailing lists with different subscribers via OUs
- Editable newsletter content that can be sent to different mailing lists
- Configurable subscription/unsubscription pages
- Administration interface to manage users and send newsletters
- Includes predefined Newsletter XML-Content



Integrate your own XML content:

- Implement the Interface `I_CmsNewsletterMailData` (extend the `A_CmsNewsletterMailData`) with methods
  - `getEmail`
  - `getEmailContentPreview`
  - `getResourceTypeName`
- Configuration via Module Parameter:
  - Own `CmsNewsletterMailData` class must be defined as module parameter `"class_maildata"`





# Alkacon OAMP Counter Module



## Key Features:

- DB-based generic counter
- Used to implement:
  - Automatic filename creation, e.g. myfile\_{x}.html where {x} is an increasing number
  - Download counters
- Administration tool to maintain counters
- API to access counters from your own Java-Code



How to use in own code:

- Access via CounterManager object

## Sample Code:

```
private String getId(String conntetId) throws CmsException {  
  
    String number = "00000";  
    // read counter from DB and increment it  
    int dbNumber = getCounterManager().incrementCounter(counterId);  
    number = number + dbNumber;  
    number = number.substring(number.length() - 5, number.length());  
  
    return number;  
}
```

Will produce a 5 digit counter number



# Alkacon OAMP Web Optimization Module



## Key Features:

- Minify, obfuscate and merge of .css and .js files
- On the fly image sprite creation

Improve website performance by

- File compression -> faster download time
- File merging -> fewer HTTP requests
- Image Sprites -> fewer HTTP requests
- Offline project always shows uncompressed original files



## Usage:

- Alkacon Optimized JS Resource (optimized.js)
  - Defines the .js files to merge
  - Compression and merge factors

## Source in JSP:

```
<%@ page session="false" import="com.alkacon.opencms.weboptimization.*" %><%  
CmsOptimizationJs oJs = new CmsOptimizationJs(pageContext, request, response);  
oJs.includeDefault("%(link.strong:optimized.js)");  
%>
```

## Online Project:

```
<script type="text/javascript" src="optimized.js" ></script>
```

## Offline Project:

```
<script type="text/javascript" src="original1.js" ></script>  
<script type="text/javascript" src="original2.js" ></script>
```

# Alkacon OAMP Syndication Feed Module



## Key Features:

- Create syndication feeds from XML contents
- Use OpenCms collectors to collect the content included in the feeds
  - Collector specifies which xml contents are contained in the feed
- Configurable mapping of XML content fields to feed fields
- Complete configuration is done in a feed XML content





# Alkacon OAMP Calendar Module



## Key Features:

- Calendar with week, month and year overviews
- Normal entries (Alkacon calendar entry)
- Serial entries (Alkacon serial date entry)
- Side elements for monthly or last recent entries
- Addition of local holidays



## Integration:

- Required Style sheet:
  - `/system/modules/com.alkacon.opencms.calendar/resources/calendar.css`
- Monthly side view:
  - Include the file  
`/system/modules/com.alkacon.opencms.calendar/elements/calendar-sidemonth.jsp`
- Last recent side view:
  - Include the file  
`/system/modules/com.alkacon.opencms.calendar/elements/calendar-sidecurrententries.jsp`
- A file of the type "Alkacon calendar view" as detail display



## Integration:

- Required Style sheet:

- `/system/modules/com.alkacon.opencms.calendar/resources/calendar.css`

- Monthly side view:

- Include the file

`/system/modules/com.alkacon.opencms.calendar/elements/calendar-sidemonth.jsp`

- Last recent side view:

- Include the file

`/system/modules/com.alkacon.opencms.calendar/elements/calendar-sidecurrententries.jsp`

- A file of the type "Alkacon calendar view" as detail display



# The Alkacon OAMP DocCenter Module



## Key Features:

- Categorized Documents
- Document lists are created by “Directory listings”
- Display of recent changes in Document center
- Full text search for documents in DocCenter
- History of older Document versions (optional)
- Disclaimer function for downloads (optional)



## Usage:

- Download areas on websites:
  - Multiple independent download areas can be stored anywhere in the VFS
- Documents centers in Intranets:
  - Example: Intranet with several thousand documents based on the OAMP DocCenter module



## Installation:

- Normal module installation
- Modification of `opencms-system.xml`:
  - Add the ResourceInit handler

```
<resourceinit>
<resourceinithandler
  class="com.alkacon.opencms.documentcenter.CmsShowDisclaimer"/>
<resourceinithandler
  class="com.alkacon.opencms.documentcenter.CmsShowDefaultFile"/>
...
</resourceinit>
```

What is a ResourceInit handler? A little excursion will follow later...





Hands-on example:

A real-live presentation explains much more than 1000 slides....



## Customize the DocCenter:

- You can write your own JSPs using the DocCenter functions for a customized layout
  - Make a copy of the original DocCenter JSP into your module
  - Customize the JSPs
  - Use your own File icons `ic_app_{extension}.gif` in the `/resources/` folder of the DocCenter module



## Configuration via Module Parameter

- Different DocCenter layouts possible for different sites

### Default:

```
defaultfile_default=/system/modules/com.alkacon.  
opencms.documentcenter/pages/documents.jsp
```

### Customized:

```
defaultfile_default=/system/modules/my.module/pages/do  
cuments.jsp
```

### Customized for special site:

```
defaultfile_mysite=/system/modules/my.other.module/pag  
es/documents.jsp
```



# Excursion: The ResourceInit Handler



## What is a ResourceInit handler ?

- It allows to read a different resource than the one shown in the URI
- “Normal” behaviour of OpenCms
  - The URI maps to a resource in the VFS
  - If the URI maps to a non existing file in the VFS, a 404-Error is thrown
- With a ResourceInit handler, we can use a “virtual” URI that does not exist in the VFS



Example:

`www.myserver.com/opencms/opencms/index.html`

is mapped to

`/sites/default/index.html`

With ResourceInit Handler the following is possible:

`www.myserver.com/opencms/opencms/special/index.html`

is mapped to

`/sites/default/internal/specialfolder/index.html`

But the URI keeps the same!



## Where is a ResourceInit handler used?

- OAMP DocCenter:
  - Directory listing
  - Disclaimer
- OpenCms Core:
  - Show history version
- From customer projects:
  - Automatic PDF Form insertion



## Implementation & Configuration

- ResourceInit Handler must implement the interface *org.opencms.main.I\_CmsResourceInit* and the method *initResource*

- Is configured in the `opencms-system.xml`:

```
<resourceinit>
```

```
...
```

```
<resourceinithandler class="com.alkacon.MyInitHandler"/>
```

```
...
```

```
</resourceinit>
```



## Implementation & Configuration

- Multiple ResourceInit handlers can be configured
- Handlers will be executed in the order of the configuration
- ResourceInit handler analyses the URI and returns a CmsResource
- Each handler should have an exclusive trigger in the URI which signals that it should be used
- Important: Each request into OpenCms will run through all ResourceInit handlers, so take care what you implement in them: **PERFORMANCE!**



## Code example:

```
public class MyInitHandler implements I_CmsResourceInit {
    /** trigger path. */
    public static final String HANDLER_PATH = "/special/";

    public CmsResource initResource(
        CmsResource resource,
        CmsObject cms,
        HttpServletRequest request,
        HttpServletResponse response) throws CmsResourceInitException {
        // only process this handler if there is a real request
        if (request != null) {
            String uri = cms.getRequestContext().getUri();
            // only process those requests that start with the trigger path
            if (uri.startsWith(HANDLER_PATH)) {
                // extract the uri of the resource we really wanted to read
                uri = uri.substring(HANDLER_PATH.length(), uri.length());
                // do your implementation here
            }
        }
        return resource;
    }
}
```

# The Alkacon OAMP Webform Module and its extensions



## Key Features:

- Customizable easy to build webforms via XML Contents
- Different types of input fields:
- Text, Text area, Radio buttons, File uploads Checkboxes etc...
- Optional Database storage
- Backoffice to manage values stored in Database
- Usage of CAPTCHA fields
- Extendable with other OAMP modules



## Customization:

- **Frontend HTML:**
  - Defined in workplace.properties file in HTML snippets
  - Copy existing file to your module and use the module parameter “message” to point to the customized file
- **Frontend CSS:**
  - Customizable class addition in Form XML-Content



## Dynamic field:

- You can define form fields that get their values from your own code
- Write a class that implements the interface `com.alkacon.opencms.formgenerator.I_CmsDynamicFieldResolver`
- Implement method  

```
public String resolveValue(CmsDynamicField field,  
CmsForm form)
```
- Define your class in the XML content of your webform



## Extensions:

- Survey module with graphical summary
- Comments module
- Those modules use the API and database provides by the Webform Module.



# The Alkacon OAMP Survey Module





## Key Features:

- Surveys based on the OAMP Webform, can be easily edited
- Entries are stored in DB
- Export of survey data for further processing
- Graphical Summary of survey results



- Only the following field types are used in the survey report:
  - Checkbox
  - Radio buttons
  - Select box
- Other fields (e.g. text boxes) are possible, but stored as individual values
- Only one participation per survey and user (Cookies)



## Customization:

- As based on OAMP Webform, the same technique is used
- **IMPORTANT:** You have to include the following .css files in your template:
  - `/system/modules/com.alkacon.opencms.survey/resources/css/webform.css`
  - `/system/modules/com.alkacon.opencms.formgenerator/resources/css/webform.css`
  - Alternatively, you can use your own customized versions of those .css files

# The Alkacon OAMP Comments Module



## Key Features:

- Comment form can be integrated in existing pages
- Moderated and unmoderated comments possible
- Different security levels
- Administration via direct edit



Integration into own pages:

- Property "comments" points to a comment configuration file
- Code inserted into template:

```
<c:set var="comments"><cms:property name="comments"
file="search"/></c:set>
<c:if test="\${!empty comments}">
  <cms:include file="% (link.weak:/system/modules/
  com.alkacon.opencms.comments/elements/comment_main.jsp)" >
  <cms:param name="cmturi" value="\${cms.requestContext.uri}" />
  </cms:include>
</c:if>
```



**Thank you very much for your attention!**

Michael Emmerich  
Alkacon Software GmbH

<http://www.alkacon.com>

<http://www.opencms.org>

