Alkacon Software GmbH

An der Wachsfabrik 13
DE - 50996 Köln (Cologne)

Geschäftsführer / CEO
Alexander Kandzior

Amtsgericht Köln
HRB 54613

Tel: +49 (0)2236 3826 - 0
Fax: +49 (0)2236 3826 - 20

http://www.alkacon.com
http://www.opencms.org

Alkacon Software GmbH

Technote

Alkacon OAMP Counter Module

Version: 1.0

Date: Monday, February 25, 2008

1 Table of Content

2 Abstract

This document describes the installation, configuration and usage of the Alkacon OpenCms Add-On Module Package Counter. With the Counter module, it is possible to manage multiple counters which can be used to count page hits or download numbers for example.

3 General purpose of the Alkacon OAMP Counter Module

The module extends a basic OpenCms installation with the capability to create highly configurable counters. It provides the following features:

- The counters can be easily managed in the administration view.
- A possible usage is to count the hits of a page or how many people have downloaded a download file.
- Another usage is the creation of unique filenames. The counter module can be used to generate a consecutive number. This number must be appended to the filename to get a unique filename. For example: filename_{x}.html *({x} is the number from the counter module)*
- A security mechanism in the administration view helps you by editing a counter in the right way.
- With the module API it is easy to include it in pages or own Java code.

4 Installation

Note: To use the Alkacon OAMP Counter module, you need at least OpenCms version 7.0.3. The module is not compatible with older OpenCms versions.

Step by step installation procedure:

1. Login to the OpenCms workplace
2. Go to the OpenCms Administration view
3. Click "Module Management" and select either "Import Module from Server" if the module was placed in the **`WEB-INF/packages/modules/`** folder of your OpenCms installation, or select "Import Module with HTTP" to upload the module from your local file system
4. Select the Alkacon OAMP Counter module zip file **`com.alkacon.opencms.counter_1.0.x.zip`** to import
5. Check if the jar file **`com.alkacon.opencms.counter.jar`** and **`com.alkacon.opencms.util.jar`** has been deployed in the **`WEB-INF/lib/`** folder after installation
6. Restart your servlet container afterwards

4.1 Configuration

In this module it's only possible to set the database pool parameter. This allows you to save the data used by the counter module in another database than the default one. If this wanted, please follow the steps under point 4.1.1.

4.1.1 Database

1. Log in to the OpenCms Workplace.
2. Switch to the Administration View.
3. Go to the Module Management page.
4. In the list of the modules click on the module name of the module **`com.alkacon.opencms.counter`**.
5. Click on Module Parameters. Adapt the setting for the module parameter **`db-pool`**.
6. Restart your servlet container afterwards

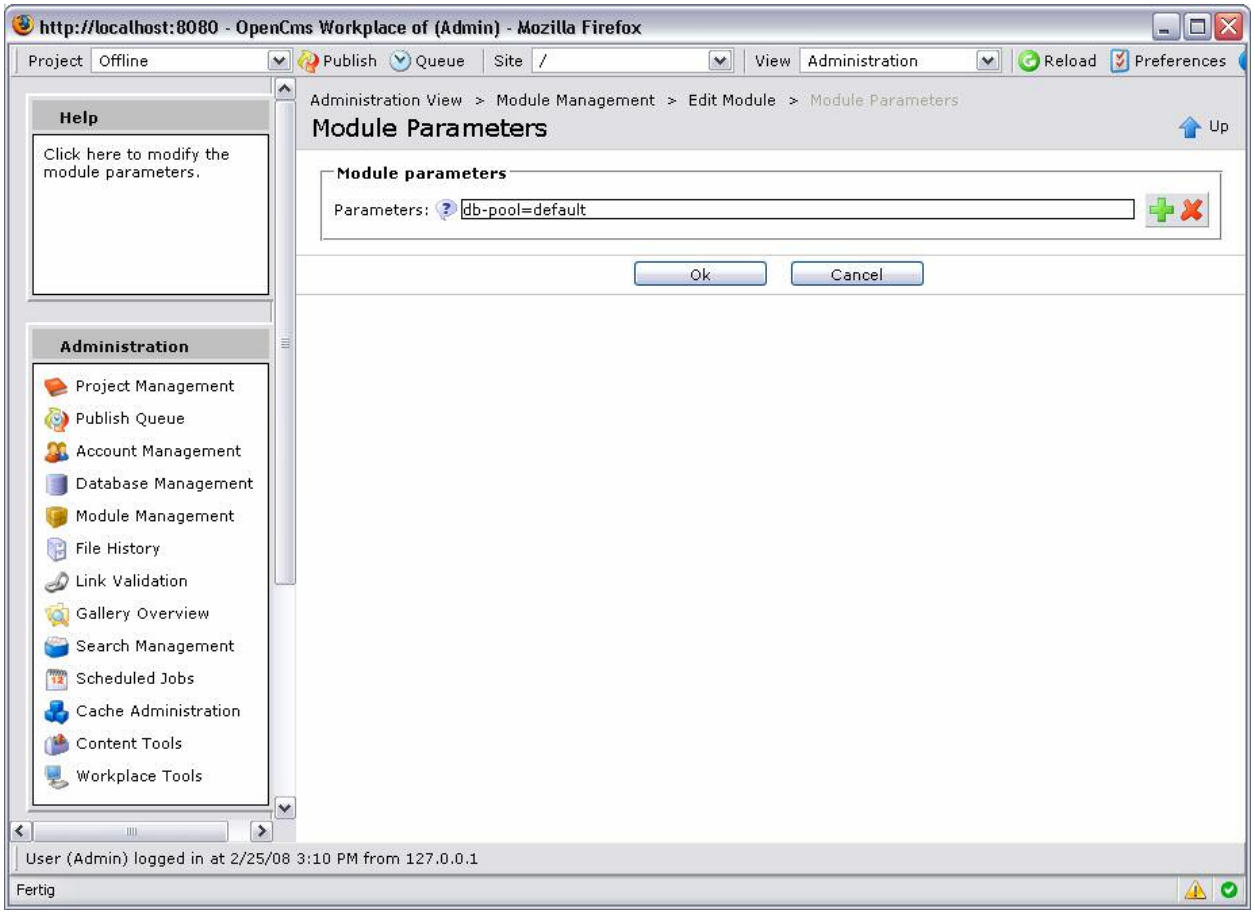

Figure 1: Module parameters of Alkacon OAMP Counter Module

The **`db-pool`** parameter value references a database pool configured in **`<tomcat-home>/webapps/<webappname>/WEB-INF/config/opencms.properties`**. The value "default" should be OK for most applications. If you need to collect your counter data in a dedicated database you may configure another pool in the opencms.properties and refer to it in this module parameter. If the parameter isn't set then at default "default" is taken.

5 Module usage

After successful installation and configuration of the Counter module, it is ready to use. It can be found in the administration view under the point “Database Management”. The icon “Counter” can be found under “Additional Tools” Box.

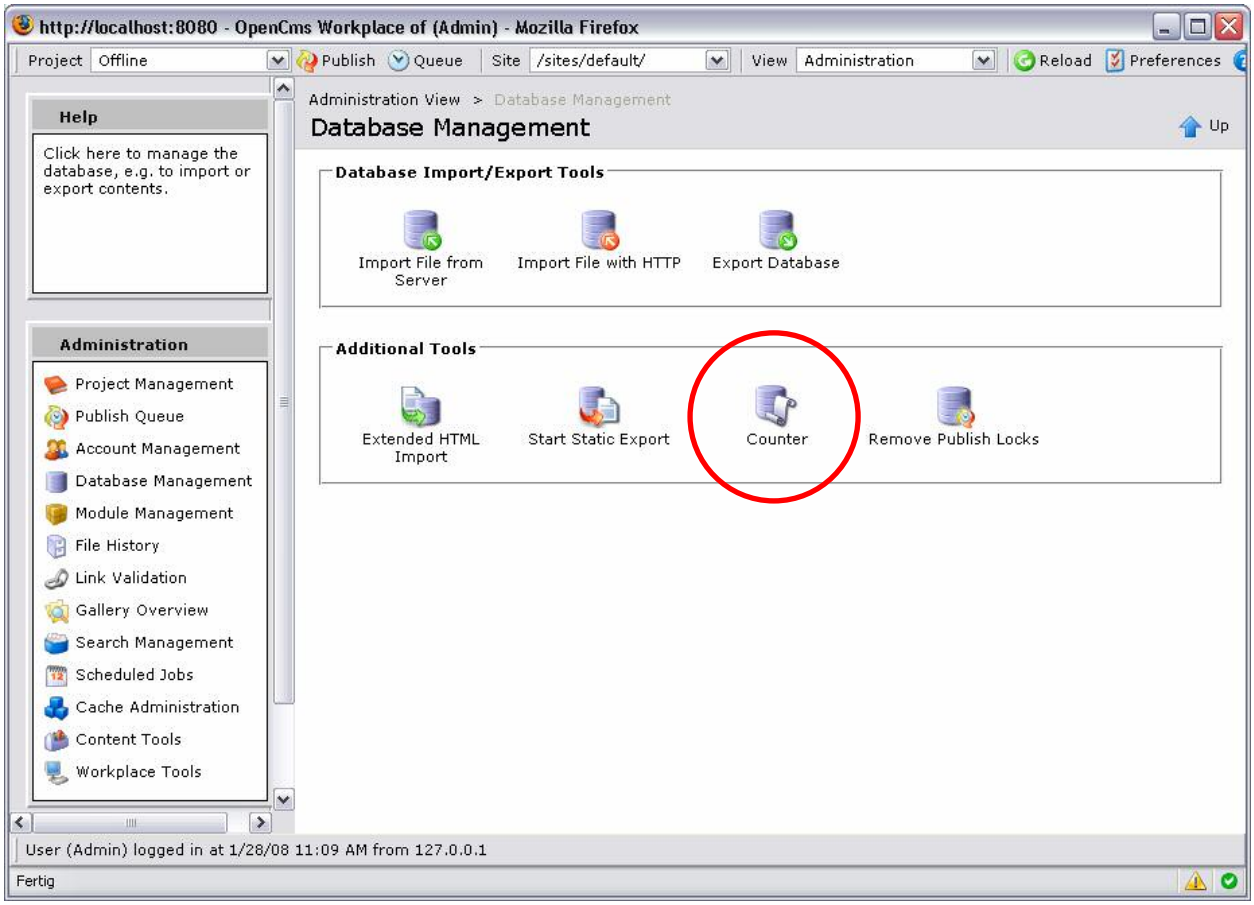

Figure 2: The administration view of „Database Management“

5.1 Management of the counters

In the management view of the counters are following fields available (see Figure 3):

- Counters: Many counters can be added. The value format is identical to the module parameters.
- Overwrite: This must be checked, if the counter entries should be updated with a lower number.

By pressing “Ok” the changes are saved persistent.

5.1.1 Counters

The format of the counters is identical to the module parameters. So it must be having following structure:

```
{key}={value}
```

For examples:

```
download=120

site hit=2000
```

The “{key}” describes the unique name of the counter and with the sign “=” the key and the value are separated. Please notice that only numbers can be entered as values.

5.1.2 Overwrite

The security option “overwrite” is needed, because at the same time when the counters are managed in the administration view, one or more of the counters could be incremented, for example by starting a new download which is monitored by a counter. Because the administration view isn’t automatically updated, the old values would be saved and the increment values would be lost. For this case the checkbox “Overwrite” is established.

If the checkbox is unselect only values which are higher, deleted counters and new created counters are saved persistent.

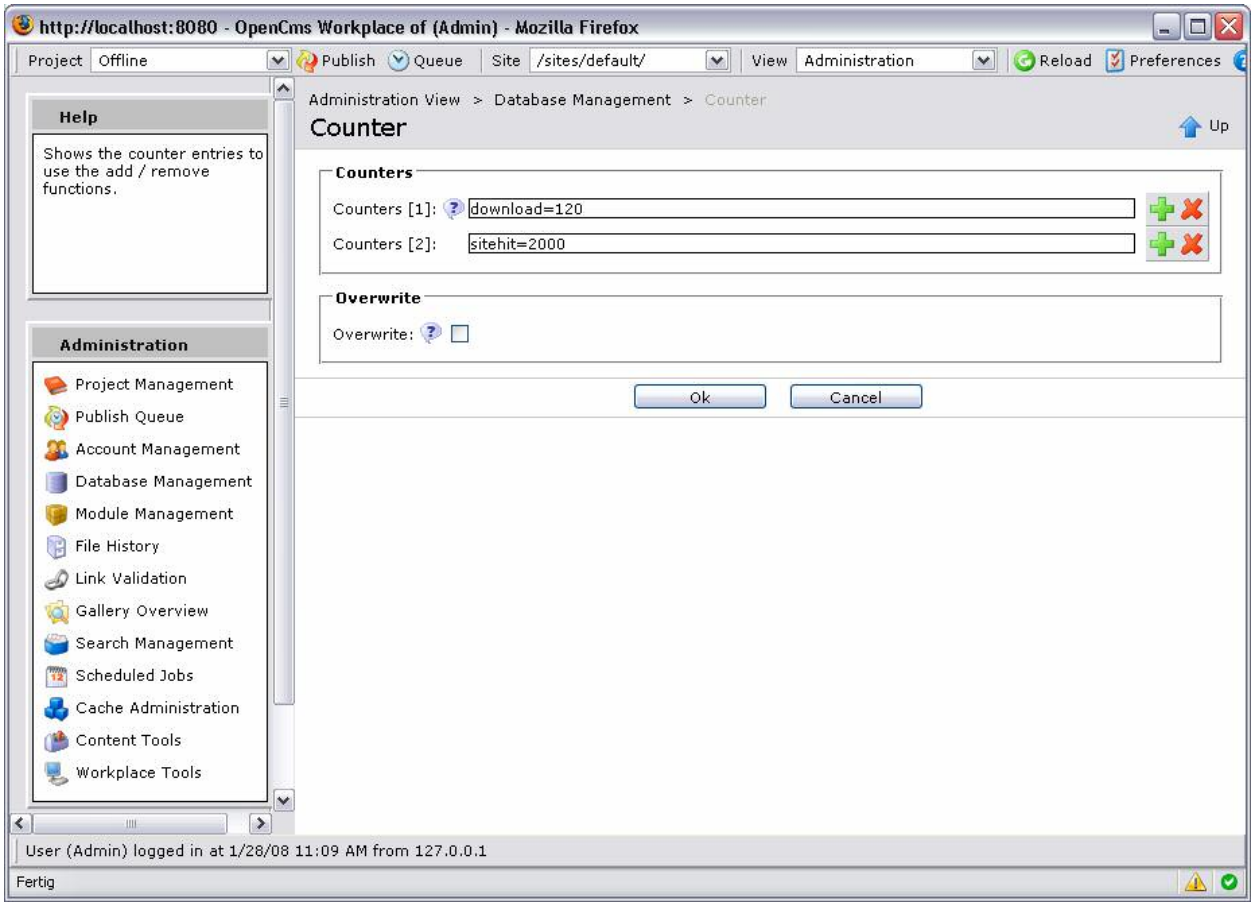

Figure 3: The management view of the counters

6 Using the module API

All classes used to generate and configure the counters are part of the package **`com.alkacon.opencms.counter`**.

This package **`com.alkacon.opencms.counter`** contains the implementation of the dialog and the access to the database. Following classes are used:

- **CmsCounterDialog**: Defines the dialog in the administration view.
- **CmsCounterManager**: Contains methods to create, delete and update a counter entry in the database.
- **CmsExampleUseBean**: Shows examples of using this module.
- **Messages**: convenience class to access the localized messages of the counter package.

7 Example of use

Following code examples shows how to use this module. All this examples are implemented in the class "`CmsExampleUseBean`" in the package **`com.alkacon.opencms.counter`**.

The first code example shows how to get the CmsCounterManager in your own code. The counter manager provides access to all counter functions and is initialized as the ActionInstance of the module:

```
    public CmsCounterManager getCounterManager() {

        CmsCounterManager result = null;
        // Get the module
        CmsModule module = OpenCms.getModuleManager().getModule(CmsCounterManager.MODULE_NAME);
        // Get the action class
        result = (CmsCounterManager)module.getActionInstance();
        if (result == null) {
            result = new CmsCounterManager();
        }
        return result;
    }
```

The following example shows how to use this module to generate a unique filename. The counter is identified by a counter key. This key is the same that is used in the administration view to display the value stored for this counter.

The function "`generateHtmlFilename()`" generates a filename of the following scheme:

FILENAME_PREFIXxxxxxFILENAME_SUFFIX

"xxxxxx" is a unique number generated by the counter module.

By executing the function "`getCounterManager().incrementCounter(counterKey);`" a new number from the database is generated and the number is automatically incremented in the database. Other methods in the "`CmsCounterManager`" class provide further additional options to manipulate the counter value.

```
public String generateHtmlFilename(String counterKey) throws CmsException {

        StringBuffer result = new StringBuffer();
        String number = "00000";
        int dbNumber = getCounterManager().incrementCounter(counterKey);
        number = number + dbNumber;
        number = number.substring(number.length() - 5, number.length());

        //append the components of the fileName
        result.append(FILENAME_PREFIX).append(number).append(FILENAME_SUFFIX);

        return result.toString();
    }
```