Built In Deployer Callbacks

CacheInvalidatePostProcessor

A callback that invalidates cache in a target system (Crafter Engine).

Sample Configuration

```
<bean id="sampleCacheInvalidate"
   class="org.craftercms.cstudio.publishing.processor.CacheInvalidatePostProcessor">
   <property name="cacheInvalidateUrl"
value="http://localhost:9090/api/1/cache/clear_all" />
</bean>
```

Configuration Properties

cacheInvalidateUrl	A cache invalidation URL

Command Processor

> ⓘ This processor is available from 2.4.4

A callback that sends runs a command line executable on Windows.

Sample Configuration

```
<bean id="sampleCommandProcessor"
                class="org.craftercms.cstudio.publishing.processor.CommandProcessor">
                <property name="command" value="testrun.exe SITE ROOT FILE STATUS" />
        </bean>
```

- SITE: replaced with site name
- ROOT: replaced with the root folder path set in a deployer target
- FILE: replaced with a file path
- STATUS: replaced with deployment status (NEW, UPDATED, DELETED)

The order of variables in a command string can be changed and also any variable can be removed.

Email Processor

> ⓘ This processor is available from 2.4.4

A callback that sends emails via SMTP based on a content deployed. The callback loads a corresponding content in preview to generate email body content.

Sample Configuration

```
<bean id="sampleEmailProcessor"
        class="org.craftercms.cstudio.publishing.processor.EmailProcessor">
        <!-- mail server properties -->
        <property name="host" value="smtp.gmail.com" />
        <property name="port" value="587" />
        <property name="protocol" value="tls" />
        <property name="authenticated" value="true" />
        <property name="username" value="mytestemail@gmail.com" />
        <property name="password" value="mypassword"/>
        <property name="defaultMailFrom" value="mytestemail@gmail.com" />
        <!-- site properties -->
        <property name="previewUrl" value="http://127.0.0.1:8080" />
        <property name="componentController" value="/crafter-controller/component" />
        <property name="headers">
   <map>
   </map>
        </property>
        <!-- content check properties -->
        <property name="matchPatterns">
   <list>
    <value>/site/website/email/.+\.xml</value>
    <value>/site/components/email/.+\.xml</value>
   </list>
        </property>
        <property name="negativeChecks">
   <list>
    <value>404</value>
   </list>
        </property>
        <property name="positiveChecks">
   <list>
    <value>Crafter News</value>
   </list>
        </property>
        <property name="urlReplacements">
   <map>
    <entry key="^/site/website"><value></value></entry>
    <entry key="\/index\.xml$"><value></value></entry>
    <entry key="\.xml$"><value>.html</value></entry>
   </map>
        </property>
        <!-- content properties -->
        <property name="mailToPath" value="//mailTo/item/email" />
        <property name="mailCcPath" value="ccList" />
        <property name="mailBccPath" value="//mailBcc/item/email" />
        <property name="replyToPath" value="//replyTo/item/email" />
        <property name="mailFromPath" value="mailFrom" />
        <property name="titlePath" value="mailTitle" />
        <property name="textContentPath" value="textContent" />
        <property name="sendEmailFlagPath" value="sendEmailFlag" />
        <property
name="imagePattern"><value>/static-assets/images/[^"]+</value></property>
        <property name="contentReplacements">
   <map>
    <entry key="http://127.0.0.1:8080"><value></value></entry>
   </map>
        </property>
    </bean>
```

Configuration Properties

host	Email server location
port	SMTP port
protocol	tls or smtps
authenticated	set true if the server requires authentication
username	username for authentication
password	password for authentication
defaultMailFrom	A default email to set as from email address. This is used when no from address is found in content
previewUrl	The preview site url
componentController	/crafter-controller/component (crafter engine 2.x and above. This is a url to render components in preview)
siteName	A site name for crafterSite parameter in preview (optional). The deployer provides this by default.
headers	A map of header key and value to add when accessing a page or a component in preview
matchPatterns	A list of regex patterns to match content path. No content will match if not set.
negativeChecks	A list of strings must not be found in the content. Email will not be sent when any of strings is found.
positiveChecks	A list of strings must be found in the content. Email will not be sent when any of strings is not found.
urlReplacements	A map of regex and replacement to transform file path to preview URL.
mailToPath	An xpath to find mailto addresses. A path containing // will be recognized as a list of values, each value containing one email address. Otherwise the field value will be treated as a semicolon-separated list. A single email address can be specified as email address only (e.g. john.doe@mailserver.com) or combined with a name (e.g. John Doe <john.doe@mailserver.com>). This property is configured with a list of values (//mailTo/item/email) in the configuration example.
mailCcPath	An xpath to find cc addresses. A path containing // will be recognized as a list of values, each value containing one email address. Otherwise the field value will be treated as a semicolon-separated list. A single email address can be specified as email address only (e.g. john.doe@mailserver.com) or combined with a name (e.g. John Doe <john.doe@mailserver.com>). This property is configured with a semicolon-separated list (ccList) in the configuration example.

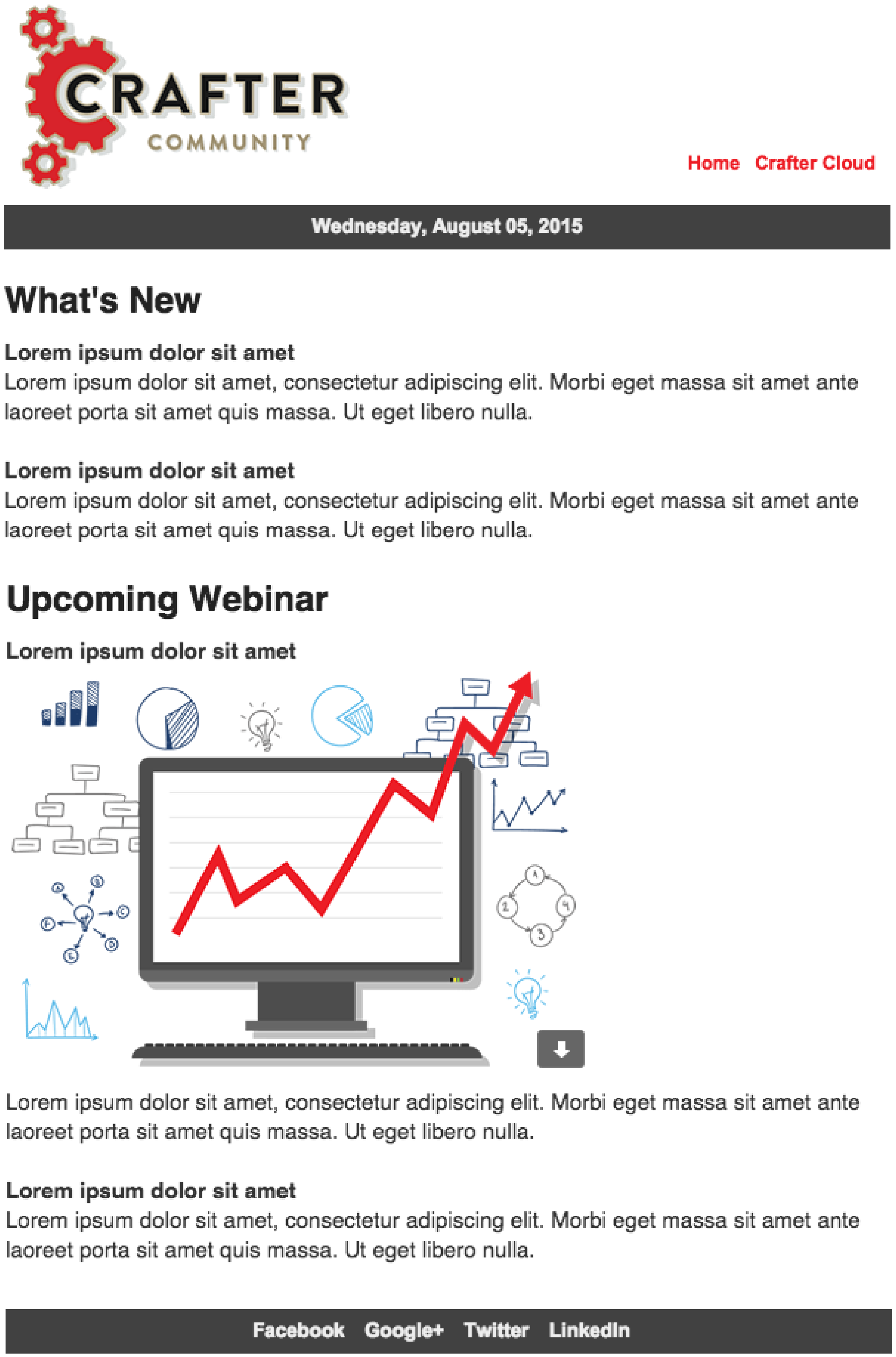
mailBccPath	An xpath to find bcc addresses. A path containing // will be recognized as a list of values, each value containing one email address. Otherwise the field value will be treated as a semicolon-separated list. A single email address can be specified as email address only (e.g. john.doe@mailserver.com) or combined with a name (e.g. John Doe <john.doe@mailserver.com>). This property is configured with a list of values (//mailBcc/item/email) in the configuration example.
replyToPath	An xpath to find reply to addresses. A path containing // will be recognized as a list of values, each value containing one email address. Otherwise the field value will be treated as a semicolon-separated list. A single email address can be specified as email address only (e.g. john.doe@mailserver.com) or combined with a name (e.g. John Doe <john.doe@mailserver.com>). This property is configured with a list of values (//replyTo/item/email) in the configuration example.
mailFromPath	An xpath to find mail from address. An email address can be specified as email address only (e.g. john.doe@mailserver.com) or combined with a name (e.g. John Doe <john.doe@mailserver.com>).
titlePath	An xpath to find email subject
textContentPath	An xpath to find email body content when no HTML is allowed
sendEmailFlagPath	An xpath to find true or false flag to send email upon content deployment or not
imagePattern	A pattern to match embedded images
contentReplacements	A map of regex and replacement to find and replace in body content. Image path in components renders with the full preview URL, so set this property to remove the preview URL from the content generated in preview as the example configuration.

Sample Content Type

News Email

1. Create 'news-email' folder in Share Repository/cstudio/config/sites/your-site/content-types/component
2. Upload the following files to the folder
   a. config.xml
   b. controller.js
   c. extract.js
   d. form-definition.xml
3. Upload the following file to Templates > web folder
   a. news-email.ftl
4. Upload the following images to Static Assets > images > email folder
   a. crafter-logo-transparent.png
   b. webinar.png
5. Create a News Email content under Components > email folder
6. Deploy the content. make sure 'Send Mail' checkbox is marked to receive an email. Also, make sure to have at least one to or cc or bcc email address.

Sample Email Output



CRAFTER COMMUNITY

Home Crafter Cloud

Wednesday, August 05, 2015

What's New

Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget massa sit amet ante laoreet porta sit amet quis massa. Ut eget libero nulla.

Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget massa sit amet ante laoreet porta sit amet quis massa. Ut eget libero nulla.

Upcoming Webinar

Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget massa sit amet ante laoreet porta sit amet quis massa. Ut eget libero nulla.

Lorem ipsum dolor sit amet
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi eget massa sit amet ante laoreet porta sit amet quis massa. Ut eget libero nulla.

Facebook Google+ Twitter LinkedIn

⚠ Images matching imagePattern with src attribute will be embedded in email using CID. However, how email clients handle CID embedded images varies especially with webmail, so there is no guarantee embedded images will work. Alternative is to use linked images from an external source, which provides the following benefits.

- Works with most email clients

- Keeps email weight light
- Can change images after sending
- Can be used to track email

SamplePostProcessor

A callback that prints all files deployed at DEBUG level

Sample Configuration

```
<bean id="sampleCacheInvalidate"
   class="org.craftercms.cstudio.publishing.processor.SamplePostProcessor"/>
```

SearchAttachmentProcessor

A callback that updates search indexes for static assets

Sample Configuration

```
<bean id="searchAttachmentProcessor"
   class="org.craftercms.cstudio.publishing.processor.SearchAttachmentProcessor">
   <property name="searchService" ref="SearchService" />
   <property name="siteName">
      <value>sample</value>
   </property>
   <property name="supportedMimeTypes">
      <list>
         <value>application/pdf</value>
      </list>
   </property>
</bean>
```

Configuration Properties

searchService	Crafter SearchService bean
siteName	A site name to override if different from what's provided from Authoring
`supportedMimeTypes`	A list of mimetypes that need to be indexed

SearchAttachmentWithExternalMetadataPostProcessor

A callback that updates search indexes with attachment metadata files

Sample Configuration

```
<bean id="SearchAttachmentWithExternalMetadataPostProcessor"

class="org.craftercms.cstudio.publishing.processor.SearchAttachmentWithExternalMetadat
aPostProcessor">
    <property name="searchService" ref="SearchService"/>
    <property name="siteName">
        <value>sample</value>
    </property>
    <property name="excludeProperties">
        <list>
            <value>binary</value>
        </list>
    </property>
    <property name="attachmentPathPatterns">
        <list>
            <value>/static-assets/attachment/([^&lt;"'\)]+)</value>
        </list>
    </property>
    <property name="metadataPathPatterns">
        <list>
            <value>/site/components/attachment/([^&lt;]+).xml</value>
        </list>
    </property>
    <property name="referenceXpathList">
        <list>
            <value>//binary/item/value</value>
        </list>
    </property>
    <property name="multivalueSeparator" value=","/>
    <property name="tokenizeAttribute" value="tokenized"/>
    <property name="tokenizeSubstitutionMap">
        <map>
            <entry key="_s" value="_t"/>
            <entry key="_smv" value="_tmv"/>
        </map>
    </property>
</bean>
```

Configuration Properties

searchService	Crafter SearchService bean
siteName	A site name to override if different from what's provided from Authoring
excludeProperties	A list of properties to exclude

attachmentPathPatterns	A list of regex to determine file path is an attachment
metadataPathPatterns	A list of regex to determine file path is a metadata content path
referenceXpathList	A list of xpaths to find attachments in content
multivalueSeparator	A separator to tokenize multiple values
tokenizeAttribute	An attribute name in content to decide whether to tokenize the corresponding field
tokenizeSubstitutionMap	A substitution map for legacy support

SearchUpdateFlattenXmlProcessor

A callback that updates indexes after merging a content with its dependencies

Sample Configuration

```
<bean id="searchFlattenXmlProcessor"

class="org.craftercms.cstudio.publishing.processor.SearchUpdateFlattenXmlProcessor">
   <property name="searchService" ref="SearchService" />
   <property name="siteName">
      <value>sample</value>
   </property>
   <property name="includeElementXPathQuery" value="//include" />
   <property name="charEncoding" value="UTF-8" />
       <property name="disableFlatteningElement" value="disableFlattening" />
       <property name="tokenizeAttribute" value="tokenized"/>
       <property name="tokenizeSubstitutionMap">
           <map>
               <entry key="_s" value="_t"/>
               <entry key="_smv" value="_tmv"/>
           </map>
       </property>
</bean>
```

Configuration Properties

searchService	Crafter SearchService bean
siteName	A site name to override if different from what's provided from Authoring
includeElementXPathQuery	An xpath to find dependencies to merge
charEncoding	Content character encoding (set to UTF-8 by default)
disableFlatteningElement	An element name in content to decide whether to merge the corresponding dependency

tokenizeAttribute	An attribute name in content to decide whether to tokenize the corresponding field
tokenizeSubstitutionMap	A substitution map for legacy support

SearchUpdateProcessor

A callback that updates search indexes

Sample Configuration

```
<bean id="searchProcessor"
   class="org.craftercms.cstudio.publishing.processor.SearchUpdateProcessor">
   <property name="searchService" ref="SearchService" />
   <property name="siteName">
      <value>sample</value>
   </property>
   <property name="charEncoding" value="UTF-8" />
       <property name="tokenizeAttribute" value="tokenized"/>
       <property name="tokenizeSubstitutionMap">
           <map>
               <entry key="_s" value="_t"/>
               <entry key="_smv" value="_tmv"/>
           </map>
       </property>
</bean>
```

Configuration Properties

searchService	Crafter SearchService bean
siteName	A site name to override if different from what's provided from Authoring
charEncoding	Content character encoding (set to UTF-8 by default)
tokenizeAttribute	An attribute name in content to decide whether to tokenize the corresponding field
tokenizeSubstitutionMap	A substitution map for legacy support

ShellProcessor

A callback that runs a command with arguments in shell.

Sample Configuration

```
<bean class="org.craftercms.cstudio.publishing.processor.ShellProcessor"
   id="md5Checksum">
   <!--  Command to Run -->
   <!--  IF asSingleCommand=true can be all in one line   -->
   <!--  Else only the actual command to execute all the args  -->
   <!-- should be in the  arguments map-->
   <property name="command" value="md5 %{files} > %{outputFile}" />
   <!-- Note This should be only set true on *nix systems -->
   <!-- where /bin/bash exist -->
   <!-- This will make  -->
   <property name="asSingleCommand" value="true"/>
   <!--  This will filter all files  -->
   <!--  If not set use all created,updated and deleted -->
   <!-- "UPDATED" only updated files -->
   <!-- "NEW" only created files -->
    <!-- "EXISTENT" created and updated files -->
   <!-- "DELETED" only delete files -->
   <property name="sourceFiles" value="EXISTENT" />
   <!--  Arguments to be append to the command -->
   <!--  the Key will be assume to be a variable in command value (value of %{}) -->
   <property name="arguments">
      <map>
         <!--  Special Argument -->
         <!--  This will create a files variable that will be use in the command
%{files} -->
         <!-- includeFilter uses java regex to match files in the changeset depending
on sourceFiles value-->
         <!-- Will generate a list (using spaces) of files -->
         <entry key="includeFilter" value=".*" />
         <entry key="outputFile" value="/tmp/md5sums.txt"/>
      </map>
   </property>
</bean>
```

Configuration Properties

command	A command to run
asSingleCommand	Run as a single command flag
sourceFiles	Set to run on which deployment type
arguments	A map of command arguments