Escenic Content Engine

Syndication Reference

5.4.7.169266


vizrt

Disclaimer

Vizrt provides this publication “as is” without warranty of any kind, either expressed or implied.

This publication may contain technical inaccuracies or typographical errors. While every precaution has been taken in the preparation of this document to ensure that it contains accurate and up-to-date information, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained in this document.

Vizrt’s policy is one of continual development, so the content of this document is periodically subject to be modified without notice. These changes will be incorporated in new editions of the publication. Vizrt may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

Vizrt may have patents or pending patent applications covering subject matters in this document. The furnishing of this document does not give you any license to these patents.

Technical Support

For technical support and the latest news of upgrades, documentation, and related products, visit the Vizrt web site at www.vizrt.com.

Last Updated

29.06.2015


vizrt

Table of Contents


vizrt

1 Introduction

This manual contains information about how to use the Escenic syndication format to import content to Escenic publications from other systems and/or export content from Escenic publications so that it can be used in other systems. The Escenic syndication format is an XML file format that can be used to represent the content and structure of any Escenic publication. It can also be used to represent **parts** of a publication: individual content items, for example.

Here is a simple example of a syndication format file representing a single content item:

```
<?xml version="1.0" encoding="utf-8"?>
<escenic xmlns="http://xmlns.escenic.com/2009/import" version="2.0">
  <content source="cnn" sourceid="1" type="news" state="published">
    <section-ref unique-name="ece_incoming" todesk="true" />
    <field name="title">Escenic Import Format</field>
    <field name="leadtext">A short and simple example.</field>
    <field name="body">
      <p>
      All text examples should contain a bit of "lorem ipsum", so:
      Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
      Nunc venenatis erat at nisl. In hac habitasse platea
      dictumst.
      </p>
      <p>
      And one more paragraph. Sed venenatis purus iaculis turpis.
      Duis sagittis luctus augue. Morbi vehicula, enim non congue
      cursus, purus dui lobortis libero, sed sagittis risus
      mauris et enim.
      </p>
    </field>
  </content>
</escenic>
```

`escenic` is the root element of all syndication format files.

`content` represents a single content item. Its **`source`** and **`source-id`** attributes together provide a unique identifier for the content item. Its **`type`** attribute specifies the content item's type as defined in an Escenic publication's **`content-type`** resource and **`state`** describes its current publication status.

`section-ref` references an existing publication section in which the content item appears. The section in question is identified by the **`unique-name`** attribute. The **`todesk`** attribute specifies that the content item appears in the section's default inbox. A **`content`** element can contain several **`section-ref`**s since a content item may appear in several sections.

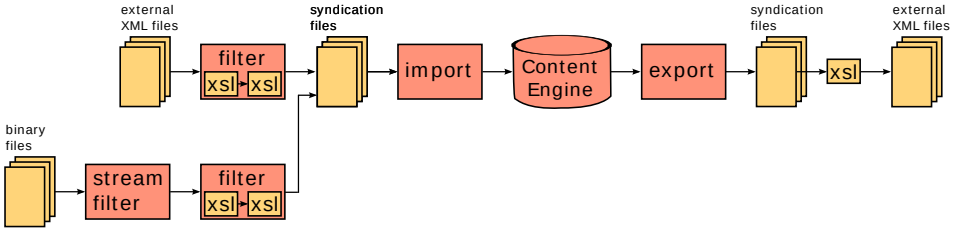
The **`field`** elements represent the content item's fields as defined by its type definition in the **`content-type`** resource. The **`"body"`** field contains a snippet of XHTML representing the main content of the content item.

This is a very simple example. A syndication file can contain many **`content`** elements. It can also contain other elements defining other publication components such as **`section`**, **`section-page`**, **`list`**, **`inbox`** or **`person`**.

For a complete description of the syndication file format see **chapter 5**. This describes all the syndication elements in detail and the combinations in which they can be used. You should read **section 1.4** before using this reference section. It contains an explanation of the conventions used in the syntax diagrams you will find there.

1.1 The Import/Export Process

The following diagram shows the flow of data when importing content into or exporting data out of an Escenic publication:



The Content Engine provides an import service that can import syndication format files, and an export service that can export syndication files. These are the central functions of the Content Engine's syndication subsystem.

However, in order to import data from another system, it is usually necessary to convert content from some external format (the source system's export format, for example) to the Escenic syndication format. Similarly, when exporting data to another system, you will usually need to convert the exported syndication files to some other format understood by the external system. The syndication subsystem therefore also provides you with a means to include such conversions as automatic steps in the import and export tasks you define.

When defining an import task, for example, you can specify a **`Filter`** through which external data is to be piped before it is submitted for import. A **`Filter`** is a chain of XSL transformations that you can define to convert data from an external XML format to Escenic syndication format. For further information, see **section 2.6**. You can also implement your own **`StreamFilter`** classes for incorporating the automated extraction of metadata from binary files such as images into the import process. For further information, see **section 2.5**.

Similarly, when defining an export task, you can define XSL transformations for converting exported syndication files to the required external format. For further information, see **section 3.1.1**.

1.2 Import

There are two different ways of importing content to the Content Engine:

- Using the import function in Web Studio to import single syndication files. This may occasionally be useful for testing or casual use, but is not recommended for production purposes. For further information, see

Escenic Content Engine Publication Administrator Guide, **section 2.5**.

- Using the **Escenic import service** (see **chapter 2**). This is the recommended approach for production purposes.

Whichever import method you use, there are a number of conditions that must be satisfied in order to successfully import content to an Escenic publication. These are discussed in the following sections.

1.2.1 Syndication File Validity

A schema defining the syndication format is delivered with the Content Engine. You will find this schema, called `syndication.rng` in the `engine/schemas` folder. You can use this schema to check the validity of any syndication files you create using any RNG-capable XML validator. You are recommended to do so at least during the development phase of an import project, since invalid syndication files are likely to result in errors in the imported content.

A valid syndication file is an important prerequisite for successful import, but no guarantee: some additional conditions must be satisfied, as described in **section 1.2.2** and **section 1.2.3**.

1.2.2 Content Identifiers

The components of an Escenic publication are inter-related in a variety of ways: content items belong to sections, sections belong to other sections, content items such as articles can contain other content items such as images, and may also contain links to other content items. Maintenance of these relations requires each content item to be uniquely identified.

Escenic import is not only intended to support one-time import of new content to a publication, but also repeated import in which existing content items are replaced by new versions: this is an important requirement for media operations where Escenic is not the primary editorial environment, but merely a means of repurposing content for the web. For this kind of repeated import it is also a requirement that the identifiers used are persistent and unchanging.

These varied requirements mean that the syndication format supports three different types of identifier. All syndication format elements that can be used to create a new object in the Escenic database may be identified in one of three ways:

- Using an `id` attribute. An `id` attribute must be unique within the current syndication file. That is the only requirement for an `id` attribute: it is not guaranteed unique in a larger context. You can use it when creating a completely new object, and you want to be able to refer to it from other elements in the same syndication file. It is not imported to the target publication.
- Using a `source` and `sourceid` attribute. These two attributes can be combined to provide a unique identifier. The `source` attribute identifies the source system and the `sourceid` is used to contain the imported

object's identifier in that system. You are recommended to use this form of identification whenever possible. It is the only type of identification that supports repeated import. Where cross-publishing is in use it may sometimes be necessary to use a **publication-name** attribute together with **source** and **sourceid** in order to indicate the publication to which a referenced content item or section belongs.

- Using a **dbid** attribute. A **dbid** attribute is intended to hold the internal Escenic database ID of an object. Such IDs can only be assigned by the Content Engine and you can therefore only use this when re-importing an existing object, and only if the object's internal ID is known. In practice this means it should only be used when re-importing an object that has previously been exported (the **dbid** attributes are always set in an exported syndication file).

An element must have at least one of the above identifiers, but may have all three. If the **dbid** attribute is set and exists in the publication, then it is used: the specified existing object is updated. None of the other attributes are imported in this case. (If you want to change the **source** and **sourceid** attributes of an existing object, then you have to supply them in an **update** element.)

If the **dbid** attribute is not set, but the **source** and **sourceid** attributes are set, then these attributes will be used to look up an existing **object**. If an object with this ID is found, then it is updated. Otherwise a new object is created and assigned the supplied **source** and **sourceid** values.

If the only ID attribute supplied is **id**, then a new object is created, with no **source** or **sourceid**. The specified **id** value is not imported.

1.2.3 References

References between the components of a publication are made using the same three identifiers described in **section 1.2.2**. If more than one of the identifiers is supplied, then they are used in the following order of priority:

dbid
This attribute is used if supplied and if the target publication contains an object with the specified **dbid**.

source and sourceid
These attributes are used if supplied and if:

- **dbid** is not supplied or cannot be used.
- the target publication contains an object with the specified **source** and **sourceid** **or** there is an element with the specified **source** and **sourceid** **before** this element in the syndication file.

id-ref
This attribute is used if supplied and if:

- **dbid** is not supplied or cannot be used.
- **source** and **sourceid** are not supplied or cannot be used.

- there is an element with the specified **`id`** **before** this element in the syndication file.

The order in which elements are imported is not significant. If an imported element references another element that has not yet been imported, the Content Engine marks the reference as unresolved. It then resolves the reference when the referenced element is imported.

1.3 Export

There are two different ways to export content from the Content Engine:

- Use the **`escenic-admin`** web application's **Export a Publication** option to export all or selected parts of a publication. For further information, see the **Escenic Content Engine Server Administration Guide**, **section 2.4.3**.
- Enable and configure the Content Engine's export service. The export service can be set up to automatically export individual content items, sections, section pages, inboxes and lists every time they are updated. Setting up such a service makes it possible to constantly export the contents of a publication as it changes, for import to some other "shadow" system that re-uses the content. For further information, see **chapter 3**.

1.4 Syntax Diagram Conventions

The reference section includes diagrams describing the syntax of the elements in a syndication file. The diagrams look something like this:

```
<content
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    state="(draft|submitted|approved|published|deleted)"?
    type="text"?
    publishdate="text"?
    activatedate="text"?
    expiredate="text"?
    publicationname="text"?
>
  <section-ref/>*
  <relation>...</relation>*
  <reference/>*
  <field>...</field>*
  <update/>?
  <author/>*
  <creator/>?
  <priority/>?
</content>
```

In these diagrams, anything appearing in plain black characters is **literal** content: that is, it should be entered in the file exactly as shown. Anything appearing in black, italic characters is a placeholder that must be replaced by something else. The only such placeholder currently in use is:

ANY

This placeholder can be replaced by any element. This is mainly intended to allow you to enter HTML elements in the body of `field` elements. You can, however, insert elements from any other namespace if you have the need.

Anything appearing in **this color** is neither literal content nor a placeholder, but has one of the following special meanings:

()
: Encloses:

  - A set of elements or attributes that are to be regarded as a group, A **?**, * or + following the closing **)** applies to the whole group.
  - A set of alternatives separated by | characters, only one of which may be selected.

|
: Separates the alternatives in a **()** set.

?
: Indicates that the preceding element is optional.

*
: Indicates that the preceding element may appear 0 or more times.

+
: Indicates that the preceding element may appear 1 or more times.

...
: Represents possible content in an element or attribute.

Element order is **not significant**. The syntax diagram shown above seems to suggest that a `sectionref` element must appear before a `relation` element, but this is **not** the case: the elements can in fact appear in any order.

---

2 The Import Service

The Content Engine has an import service that can be configured to automatically run a number of independent import tasks. For each defined task, the import service periodically checks a specified folder on the server for new files to import. If files are available, they are imported as specified in the task configuration file and archived in another folder.

How many different import tasks you define is up to you. You will need to create at least one task for each publication to which you want to import content, but you may find it useful to create more than one task per publication. You might, for example, need separate tasks for importing content from different sources.

You can set up the import service to generate reports containing information about the import tasks it performs. For more information about this, see **section 2.3**.

The import service runs all defined import tasks once a minute.

2.1 Enabling The Import Service

In order to enable the import service, you need to open the file *configuration-root*`/Initial.properties` for editing in one of your configuration layers, and uncomment the following line (at the bottom of the file):

```
service.9-xml-import-service=/com/escenic/syndication/xml/XMLImportSchedule
```

2.2 Creating Import Tasks

To create an import task:

1. Create a folder for the import task on the server. You are recommended to create an `import` folder under your *configuration-root*`/com/escenic/syndication/xml` folder, with subfolders for each import task. Each import task folder **may** have a subfolder for filters. For example:

   configuration-root`/com/escenic/syndication/xml`
   ```
   import
     pub1
       [filters]
     pub2
       [filters]
   ```

   This is the folder structure on which subsequent examples and discussions are based. You are free, however, to organize your import configuration files in any way you choose.

2. Copy a configuration file into the import task folder. You will find a template configuration file called **`XMLImportConfiguration.properties`** in the **`/engine/contrib/import`** folder.
3. Rename the configuration file to the same name as the task folder.
4. Edit the configuration file to meet your requirements. See **section 2.2.1.1** for detailed instructions.
5. Add a reference to the new import task to the **`importConfigurations`** property in the *configuration-root*`/com/escenic/syndication/xml/XMLImportService.properties` file. See **section 2.2.1** for detailed instructions.

The above steps are sufficient for the simplest case where you have "ready to import" content that is already converted to Escenic syndication format. In more complex cases you may also need to create additional **filters** and or **prefilters** in order to convert the source data to syndication format before import.

2.2.1 Editing XMLImportServices.properties

This configuration file (located in the *configuration-root*`/com/escenic/syndication/xml` folder) contains a property called **`importConfigurations`**, that references the configuration files for all the import tasks that the import service is to run. The configuration file references must:

- **Include** a relative path from the *configuration-root* folder.
- **Exclude** the **`.properties`** extension.
- Be separated by commas.

For example:

```
importConfigurations=./import/pub1/pub1,./import/pub2/pub2
```

2.2.1.1 Editing Task Configuration Files

An import configuration file contains definitions of the following properties:

`importDirectory`
The path of an import folder on the server that the import service will watch for new files. Any files appearing in this folder that match the specifications given with the **`filenames`** property will be automatically imported. For example:

```
importDirectory=/var/spool/escenic/import/pub1
```

`fileNames`
The files to watch for in the import folder specified with **`importDirectory`**. You can use wildcards, and you can specify several file names, separated by commas. You can also specify files in other folders by preceding the file names with relative paths. For example:

```
fileNames=*.xml,feeds/ntb.rss,feeds/reuters.xml
```

archiveDirectory
The path of an archive folder on the server to which successfully imported files will be copied. The folder is created if it does not exist. Here is an example **archiveDirectory** specification:

```
archiveDirectory=/var/cache/escenic/import/pub1/archive
```

errorDirectory
The path of an archive folder on the server to which files that fail to import will be copied. The folder is created if it does not exist. Here is an example **errorDirectory** specification:

```
errorDirectory=/var/cache/escenic/import/pub1/errors
```

contentDirectory
The path of an archive folder on the server in which an individual syndication file will be stored for each content item processed. The folder must contain three sub-folders called **success**, **warning** and **error**. Each of these three directories again has a sub-folder for each content type imported.

For every content item that is successfully imported, a syndication file is stored in the **success/content_***content-type* folder.

For every content item that is imported with warnings, a syndication file is stored in the **warning/content_***content-type* folder, with the warnings embedded as comments.

For every content item that fails to be imported, a syndication file is stored in the **error/content_***content-type* folder, with the error messages embedded as comments. The files are given names of the form:

source **_** *source-id* **.xml**

contentDirectory and its sub-folders are created if they do not exist.

You will most likely find it more useful to define a **contentDirectory** than an **archiveDirectory** and **errorDirectory**. It is easier to trace errors using **contentDirectory** output, since error messages are packaged together with the content item that caused them. You can, however, define all three folders if you wish.

Here is an example **contentDirectory** specification:

```
contentDirectory=/var/cache/escenic/import/pub1/content
```

publicationId
The id of the publication to which files are to be imported. You can use **publicationName** instead of this property to identify the publication if you wish. If you specify both **publicationId** and **publicationName** then **publicationId** is used. For example:

```
publicationId=pub1
```

publicationName
The name of the publication to which files are to be imported. You can use **publicationId** instead of this property to identify the publication if you wish. If you specify both **publicationId** and **publicationName** then **publicationId** is used. For example:

```
publicationName=pub1
```

defaultSectionId
The id of a default section for imported content items. Content items that are not explicitly assigned to sections in the imported syndication file are added to this section. You can use **defaultSectionName** instead of this property to identify the default section if you wish. If you specify both **defaultSectionId** and **defaultSectionName** then **defaultSectionId** is used. For example:

```
defaultSectionId=sports
```

defaultSectionName
The name of a default section for imported content items. Content items that are not explicitly assigned to sections in the imported syndication file are added to this section. You can use **defaultSectionName** instead of this property to identify the default section if you wish. If you specify both **defaultSectionId** and **defaultSectionName** then **defaultSectionId** is used. When using **defaultSectionName** to look up the default section it is first matched against unique section names. If no match is found, then it is matched against section names. For example:

```
defaultSectionName=Sports and Leisure
```

defaultUserId
The id of a default user for imported content items. The specified user is set as the creator of imported content items that do not have an explicitly specified creator in the imported syndication file. You can use **defaultUserName** instead of this property to identify the default section if you wish. If you specify both **defaultUserId** and **defaultUserName** then **defaultUserId** is used. For example:

```
defaultUserId=smithj
```

defaultUserName
The name of a default user for imported content items. The specified user is set as the creator of imported content items that do not have an explicitly specified creator in the imported syndication file.. You can use **defaultUserId** instead of this property to identify the default section if you wish. If you specify both **defaultUserId** and **defaultUserName** then **defaultUserId** is used. For example:

```
defaultUserName=John Smith
```

fileComparatorClass
The name of a Java class that is used to determine the order in which files found in the import folder are imported. For example:

```
fileComparatorClass=com.escenic.syndication.xml.comparator.DateComparator
```

The following predefined classes are provided:

`com.escenic.syndication.xml.comparator.DateComparator`
Files are imported in date order (oldest first). The date used for comparison purposes is the "last edit" date normally shown in file listings, not the creation date.

`com.escenic.syndication.xml.comparator.DateComparatorDesc`
Files are imported in reverse date order (newest first). The date used for comparison purposes is the "last edit" date normally shown in file listings, not the creation date.

`com.escenic.syndication.xml.comparator.NameComparator`
Files are imported in alphabetical order.

`com.escenic.syndication.xml.comparator.NameComparatorDesc`
Files are imported in reverse alphabetical order.

You can add other sorting methods by implementing your own comparator classes that implement the **`java.util.Comparator`** interface. The import service supplies two File objects as parameters to the comparator class's **`compare()`** method.

If you do not specify a comparator, then files are imported in alphabetical order.

`preFilters`
This property can be used to specify prefilters for processing input files. It is a **mapped** property. This means you can assign multiple indexed values to the property using dotted indices as follows:

```
preFilters.jpg=./filter/IPTCStreamFilter
preFilters.default=./filter/DefaultPreFilter
```

The property index is used as a file type selector: the first example above specifies that all import files with the extension **`.jpg`** are to be processed by **`filter/IPTCStreamFilter`** (see **section 2.5**). The special index **`default`** can be used to specify a default prefilter that will be used to process all files that are not specifically selected. If you do not specify a default prefilter, then any input files that are not specifically selected will not be processed by a prefilter. The referenced prefilters are actually **`.properties`** files in the **`filter`** subfolder containing prefilter configuration properties. For the above example the **`filter`** subfolder would need to contain two files called **`IPTCStreamFilter.properties`** and **`DefaultPreFilter.properties`**. For information about prefilter configuration, see **section 2.5**.

You can specify several prefilters, separated by commas. The specified filters then form a chain in which output from one is passed as input to the next. For example:

```
preFilters.jpg=./filter/IPTCStreamFilter1,./filter/IPTCStreamFilter2
```

SAXFilters
This property can be used to specify filters for processing input files. It is a **mapped** property. This means you can assign multiple indexed values to the property using dotted indices as follows:

```
SAXFilters.jpg=./filter/DebugFilter,./filter/IPTCFilter,./filter/DebugFilter
SAXFilters.default=./filter/DebugFilter,./filter/DefaultFilter,./filter/DebugFilter
```

The property index is used as a file type selector: the first example above specifies that all import files with the extension **.jpg** are to be processed by a particular chain of filters. The special index **default** can be used to specify a default filter chain that will be used to process all files that are not specifically selected. If you do not specify a default filter chain, then any input files that are not specifically selected will not be filtered. The referenced filters are actually **.properties** files in the **filters** subfolder containing filter configuration properties. For the above example the **filters** subfolder would need to contain two files called **IPTCFilter.properties** and **DefaultFilter.properties**. For information about filter configuration, see **section 2.6**.

importReporter
Specifies the location of the import reporter configuration file.

```
importReporter=/com/escenic/syndication/xml/ImportReporter
```

For more information about configuring the import reporter see **section 2.3**.

2.3 Configuring an Import Reporter

To configure an **ImportReporter** you must edit the supplied **ImportReporter.properties** file or create a **.properties** file of your own. It may contain the following property settings:

$class
This property must be set to **com.escenic.syndication.xml.ImportReporter**.

emailSender
Specify the location of a **MailSender** configuration file. The default value is

```
emailSender=/neo/io/services/MailSender
```

enabled
If set to **false** then no output will be generated.

reportConfiguration
With this property you can specify the location of one or more report configuration files, each of which defines in detail the content of an import report. It is a mapped property, so you can configure different report contents according to the outcome of import tasks. For Example:

```
reportConfiguration.SUCCESS=./SuccessReportConfiguration
```

```
reportConfiguration.DEFERRED=./DeferredReportConfiguration
reportConfiguration.WARNING=./WarningReportConfiguration
reportConfiguration.FAILURE=./FailureReportConfiguration
```

For more information about configuring reporter, see **section 2.4**.

2.4 Configuring Reports

You will need to create a **`.properties`** file for each of the report types you specify with the **`reportConfiguration`** property in your **`ImportReporter.properties`** file. With these **`.properties`** files you can specify in detail what your import reports should contain.

`$class`
This property must be set to **`$class=com.escenic.syndication.xml.filter.ReportConfiguration`**.

`includeStackTrace`
Specifies whether or not to include stack trace in the report.

```
includeStackTrace=true
```

`includeStatistics`
Specifies whether or not to include statistics in the report.

```
includeStatistics=true
```

`boilerplate`
Boilerplate text output on the first line of the report.

`filename`
The full path of the file to which the report is to be written. If the file does not exist then it will be created. If the file does exist, then the report will be appended to the existing contents of the file.

```
filename=/var/log/escenic/import-report.txt
```

`recipients`
A comma-separated list of email address to which the report is to be sent. The addresses must conform to RFC822 syntax. For example:

```
recipients=user@domain.com
```

`subject`
The subject line of the report email. For example

```
subject=XML Import Error Report
```

`includeAttachment`
Specifies whether or not to attach the import file(s) with the report email.

```
includeAttachment=true
```

maxAttachmentSize
The maximum allowed size for attachments, specified in kilobytes. If the import file exceeds this limit then it will not be attached with the report. If you do not want to set a size limit, set the value to **-1**.

enabled
Set to **false** to disable reporting.

2.5 Configuring Prefilters

A prefilter is a Java class that can read a binary object and generate a corresponding XML file containing sufficient information to enable the object to be imported to a publication.

Only one standard prefilter is supplied with the Content Engine:

IPTCStreamFilter
This filter makes it possible to easily import JPEG images to Escenic publications without needing to explicitly create syndication files for them, as long as the JPEG files contain IPTC metadata. The filter reads JPEG files and generates XML files containing the metadata it finds. These XML metadata files can then be converted to corresponding syndication files by a suitable filter (see **section 2.6** for more information about this).

You can make your own prefilters by writing Java classes that implement the interface **com.escenic.syndication.xml.filter.StreamFilter**. For more information about this, see **section 2.7**.

To configure a prefilter you must create a **.properties** file for it. It needs to contain only one line, specifying the fully-qualified name of the required prefilter class.

2.5.1 IPTCStreamfilter Configuration

Configuring the supplied **IPTCPStreamFilter** is slightly more complicated than configuring a standard filter. The reason for this is that it is actually implemented as a wrapper around a legacy class, **com.escenic.syndication.xml.filter.IPTCPreFilter**. This means that you actually need to create two **.properties** file, one for **IPTCPStreamFilter** and one for **IPTCPreFilter**. However, examples of all the files you need are supplied in the **/engine/contrib/import/filter** folder, so you can just copy them to an appropriate location in one of your import task configuration folders and set the **preFilters.jpg** property of the task configuration file to point to **IPTCStreamFilter.properties**.

IPTCStreamFilter.properties must contain the following lines:

```
$class=com.escenic.syndication.xml.filter.LegacyFilterSupport
filter=./IPTCPrefilter
```

IPTCPreFilter.properties may contain the following lines:

```
$class=com.escenic.syndication.xml.filter.IPTCPreFilter
imageEncoding=ISO-8859-1
```

The **imageEncoding** property is optional. It specifies the encoding of the IPTC data embedded in the images. It may be set to either **ISO-8859-1** or **UTF-8** (the default).

Given the example folder structure described in **section 2.2**, if you wanted to use the filter for the publication **pub1** then you would copy these files to *configuration-root*/**import/pub1/filters**. and set **preFilters.jpg** in *configuration-root*/**import/pub1/ImportConfiguration.properties** to **./filters/IPTCStreamFilter**.

2.6 Configuring Filters

Filters are SAX filters (that is, they are based on the XML parser called SAX) and are intended to support the conversion of XML data to Escenic syndication files suitable for import to the Content Engine.

The following standard filters are supplied with the Content Engine:

com.escenic.syndication.xml.filter.XSLFilter
: This filter applies a sequence of one or more XSL transformations to import files.

com.escenic.syndication.xml.filter.DebugFilter
: As its name suggests, this filter is purely for debug purposes. It writes the content of the input stream to a file or to standard output. You can insert it before and after **XSLFilter** transformations during filter development in order to see the effect of the transformations.

You are not recommended to attempt creation of your own filter classes. You can create custom filters by writing XSL transformations. For more information about this, see **section 2.8**.

2.6.1 XSLFilter Configuration

To configure an **XSLFilter** you must create a **.properties** file for it. It may contain the following property settings:

$class
: This property must be set to **com.escenic.syndication.xml.filter.XSLFilter**.

filter
: This must specify the relative paths of one or more XSL transformations to be run on the input data, separated by commas. The transformations are run in sequence: output from one transformation is piped into the following transformation. For example:

```
filter=./transformation1.xsl,./transformation2.xsl
```

cacheFilter
If set to **false**, then XSL transformations are not cached, but reloaded every time they are used. The default setting is **true**: transformations are only loaded once and cached for future use. It can be useful to set **cacheFilter** to **false** during development, so that changes to the transformations are used without needing to explicitly reload them.

If, for example, you have created a transformation called **iptc.xsl** for converting the XML output by **IPTCStreamFilter** to syndication format, then to make use of the transformation, you could create an **XSLFilter** configuration called **IPTCFilter.properties** with the following content:

```
$class=com.escenic.syndication.xml.filter.XSLFilter
filter=./iptc.xsl
```

Given the example folder structure described in **section 2.2**, if you wanted to use the filter for the publication **pub1** then you would place the file in *configuration-root*/**import/pub1/filters**, and set **SAXFilters.jpg** in *configuration-root*/**import/pub1/ImportConfiguration.properties** as follows:

```
SAXFilters.jpg=./filter/IPTCFilter
```

or (if you need to be able to see the input and output for debugging purposes):

```
SAXFilters.jpg=./filter/DebugFilter,./filter/IPTCFilter,./filter/DebugFilter
```

You will find a sample **XSLFilter** properties file called **XSLFilter.properties** in the **/engine/contrib/import/filter** folder.

2.6.2 DebugFilter Configuration

To configure a **DebugFilter** you must create a **.properties** file for it. It may contain the following property settings:

$class
This property must be set to **com.escenic.syndication.xml.filter.DebugFilter**.

filename
This must specify the path of the file to which output should be directed. For example:

```
filename=/var/log/escenic/import/pub1/xmldebug.log
```

If you include the string **%c** in the path it will be substituted by a counter that is incremented each time the debug filter is run. This, for example:

```
filename=/var/log/escenic/import/pub1/xmldebug_%c.log
```

will produce a series of files in **/tmp/import/pub1** called **xmldebug_1.log**, **xmldebug_2.log**, etc.

You can also specify **stdout** or **stderror** in order to direct output to the console. For example:

```
filename=stdout
```

If, for example, you have created a transformation called **`iptc.xsl`** for converting the XML output by **`IPTCStreamFilter`** to syndication format, then to make use of the transformation, you could create an **`XSLFilter`** configuration called **`IPTCFilter.properties`** with the following content:

```
$class=com.escenic.syndication.xml.filter.XSLFilter
filter=./iptc.xsl
```

Given the example folder structure described in **section 2.2**, if you wanted to use the filter for the publication **`pub1`** then you would place the file in *configuration-root*`/import/pub1/filters`, and set **`filter.jpg`** in *configuration-root*`/import/pub1/ImportConfiguration.properties` as follows:

```
SAXFilters.jpg=./filter/IPTCFilter
```

or (if you need to be able to see the input and output for debugging purposes):

```
SAXFilters.jpg=./filter/DebugFilter,./filter/IPTCFilter,./filter/DebugFilter
```

You will find a sample **`Debug`** properties file called **`XSLFilter.properties`** in the **`/engine/contrib/import/filter`** folder.

2.7 Making a Custom PreFilter

To make a custom prefilter you must create a Java class that implements the following interface:

```
package com.escenic.syndication.xml.filter;
import java.io.InputStream;

public interface StreamFilter {
  InputStream filter(InputStream pStream);
}
```

Prefilters can be chained, so the **`filter()`** method accepts a data stream that is either the content of the original file to be imported or the output of the preceding prefilter in the chain. The last prefilter in a chain must output well-formed XML.

A prefilter may throw an exception if it encounters an error or unexpected data in the input stream. This will cause the import service to abandon processing of the current import file and move the original file to the error directory specified in the task configuration (see **section 2.2**).

2.8 Making XSL Transformations

The source files to be imported by an import task are frequently XML files that have been exported from some external system. It is commonly the case, therefore that you need to convert input files from the export format of a foreign system to Escenic syndication format before it can be imported. It may also be the case that you need to generate syndication format files from XML files output by a prefilter.

Both of these cases can be solved by writing an XSL transformation or a series of XSL transformations to be executed in a chain.

Note that `IPTCStreamFilter` is a generic filter: it simply outputs all the IPTC attributes it finds in a JPEG file as well-formed XML. The actual IPTC attributes found in JPEG files are very variable, as is the way in which the attributes are used. `iptc.xsl`, however, is not generic, and will only work for JPEG that contain a specific set of IPTC attributes. In order to be used for production purposes it must be modified to handle the specific types of IPTC data occurring in the production environment.

3 The Export Service

To enable continuous export of content from an Escenic publication, you must create an **export task**. An export task defines how content items and sections are to be exported from a single publication. You must therefore create at least one export task for each publication that you want to export from. You may find it useful to create more than one task per publication. You might, for example, need separate tasks for exporting content to different target formats.

Sections and content items are always exported as individual XML files called either **`section`***id*`.xml`** or **`article`***id*`.xml`**, where *id* is the internal ID of the exported section or content item. By default, the exported files are output in Escenic syndication format. You can, however, set up an export task to apply an XSL transformation to the output data stream, in order to convert the exported items to a different format.

The export service is intended to support **continuous** export of a publication's content. Content items, sections and so on are monitored, and exported every time they are modified. If you just want to carry out a one-time export of a whole publication or part of a publication, then you should use the **`escenic-admin`** web application's **Export a Publication** option (see the **Escenic Content Engine Server Administration Guide**, **section 2.4.3**).

3.1 Creating Export Tasks

To create export tasks:

1. Add the following to your **`Initial.properties`**

   ```
   service.0.98-exportManager=/com/escenic/syndication/xml/ExportManager
   ```

   This will enable your export service
2. Create a text file called **`ExportManager.properties`** in your *configuration-root*`/com/escenic/syndication/xml` folder.
3. For each export task you want to create, add a configuration file to the *configuration-root*`/com/escenic/syndication/xml` folder. You will find a template configuration file called **`ExportConfiguration.properties`** in *configuration-root*`/com/escenic/syndication/xml`. Rename each file appropriately (for example, **`Pub1ExportConfiguration.properties`**, **`Pub2ExportConfiguration.properties`** and **`Pub3ExportConfiguration.properties`**).
4. For each export task, add a line to your **`ExportManager.properties`** file referencing the corresponding configuration file. For example:

   ```
   configuration.pub1=./Pub1ExportConfiguration
   configuration.pub2=./Pub2ExportConfiguration
   configuration.pub3=./Pub3ExportConfiguration
   ```

5. Edit each configuration file to meet your requirements. See **section 3.1.1** for detailed instructions.

3.1.1 Export Task Configuration Files

An export configuration file contains definitions of the following properties:

`name`
The name of this export configuration. This property is optional, but is useful for documentation and debugging purposes: it is included in log messages.

`objectTypes`
The object types to be exported. You can specify any combination of the following keywords, separated by commas:

`article`
Export all content items.

`section`
Export all sections.

`pool`
Export all section pages, inboxes and lists.

Alternatively, you can specify **`*`**, which means export everything: this is the default.

`sections`
The sections (including all descendant sections) from which content may be exported, specified by section name. The section names must be separated by commas. The special value **`*`** means "all sections". You can use **`sectionsById`** instead of this property. If neither this property nor **`sectionsById`** is specified, then all sections are exported.

`sectionsById`
The sections from which content may be exported, specified by ID. The section IDs must be separated by commas. You can use **`sections`** instead of this property. If neither this property nor **`sections`** is specified, then all sections are exported.

`enabled`
Enables this export task when set to **`true`**. The default is **`false`**.

`disableEvents`
Not currently used.

`includeReferredObjects`
When set to **`true`**, any content items that are referenced in an exported content item are also exported. When set to **`false`**, this is not done.

templateDirectory
The absolute path of a folder on the server. If specified, the folder must contain XSL transformations to be applied to any exported sections or content items. The transformation to be applied to sections must be called **section.xsl**, and the transformation to be applied to content items must be called **article.xsl**.

If this property is not specified, then no transformation is applied, and exported sections/content items are output as Escenic syndication files. If it is specified and the specified folder contains correctly named XSL transformations, then exported data is piped to these transformations, and the saved export files will contain the results of the transformations.

cache
If set to **true**, then the XSL transformation in the **templateDirectory** (if specified) is cached. If set to **false** (the default), then it is not.

import
If **disableEvents** is set to **false**, then every time a section or content item is **imported**, it will also automatically be **exported**. You can prevent this happening by setting this property to **false** (the default). If you actually want imported sections/content items to be exported again, set it to **true**.

exportTarget
The folder (on the server) in which exported files are to be saved, specified as an absolute file URL (**file:///tmp/escenic/export/pub1**, for example). You can include system properties in the URL (**file:///${java.io.tmpdir}/export/pub1**, for example). The default if this property is not specified is **file:///${java.io.tmpdir}**.

exportWriter
The **ExportWriter** implementation that is used to carry out the export process. You can replace this default **ExportWriter** (which outputs Escenic syndication format) with your own implementation if you need to be able to export to some other format. See **section 3.1.2** for for further information.

3.1.2 Creating your own ExportWriter

If you need to export content directly to a format other than Escenic syndication format, you can do so by writing your own **ExportWriter** implementation. You might, for example want to be able to export content directly to a format that can be read by another vendor's web service.

To do this you need to:

1. Create a component that implements the **com.escenic.syndication.xml.ExportWriter** interface.
2. Create an export task configuration file in which the **exportWriter** property points to your new component.

The interface you need to implement looks like this:

```
package com.escenic.syndication.xml;

import neo.xredsys.api.IOObject;

public interface ExportWriter {

  /**
    * Write the object using the given export configuration.
    * This method should only be called once per stream.
    *
    * @param pObject the object to export
    * @param pConfiguration the configuration to use.
    */
  void writeObject(final IOObject pObject, final ExportConfiguration pConfiguration);
}
```

You can implement the above interface directly if you wish, but you are recommended to use the abstract class **`com.escenic.syndication.xml.AbstractExportWriter`**.

3.1.3 Example Implementation

The following example shows a custom **`ExportWriter`** implementation based on **`com.escenic.syndication.xml.AbstractExportWriter`**. It is an extremely simplified example that just sends the generated XML to a specified URI:

```
package com.my.company.syndication;

import java.io.IOException;
import java.io.OutputStream;
import java.net.URI;

import neo.xredsys.api.IOObject;

public class MyExportWriter extends AbstractExportWriter {

  public MyExportWriter() {
    //By design
  }

  @Override
  protected OutputStream getOutputStream(final IOObject pObject, final URI pBaseURI) throws
 IOException {
    URLConnection connection = pBaseURI.toURL().openConnection();
    connection.setDoOutput(true);
    return connection.getOutputStream();
  }
}
```

To be able to use this **`ExportWriter`** implementation you would need to:

1. Create a configuration file for your component. This file must be added to one of your configuration layers, in the location *configuration-root*`/com/my/company/syndication/MyExportWriter.properties`. It must contain at least the following line:

   ```
   $class=com.my.company.syndication.MyExportWriter
   ```

   For more information about configuration layers, see **Escenic Content Engine Server Administration Guide**, **chapter 4**.
2. Create an export task configuration file (see **section 3.1.1**) in which the **`exportWriter`** property is set to point to your **`ExportWriter`** implementation's configuration file. For example:

   ```
   exportWriter=/com/my/company/syndication/MyExportWriter
   ```

You can find further information in the **`ExportWriter`** interface's JavaDoc.

4 Tips and Tricks

This chapter contains advice on how to carry out some of the more difficult syndication-related tasks: things you are unlikely to be able to find out from reading the reference section alone. It is currently very short, but is expected to grow.

4.1 Importing Soft Crop Definitions

Image content items can contain soft-crop definitions: image cropping co-ordinates that allow the Content Engine to generate a variety of cropped versions of an image on the fly, rather than having to store multiple versions of every image. In order for a publication to support soft-cropping of images, its image content types (as defined in the **`content-type`** resource) must include a field for storing this information. Here is an example of a content type definition that includes such a field, called **`representations`** in this case:

```
<content-type name="image">
...
  <panel name="crop">
    <ui:label>Cropped Versions</ui:label>
    <field mime-type="application/json" type="basic" name="representations">
      <ui:label>Image Versions</ui:label>
      <rep:representations type="image-versions">
        <rep:representation name="thumbnail">
          <rep:output width="100" height="100"/>
          <rep:crop/>
          <rep:resize/>
        </rep:representation>
        <rep:representation name="narrow">
          <rep:output width="500" height="400"/>
          <rep:crop/>
          <rep:resize/>
        </rep:representation>
        <rep:representation name="wide">
          <rep:output width="1000" height="800"/>
          <rep:crop/>
          <rep:resize/>
        </rep:representation>
      </rep:representations>
    </field>
  </panel>
...
</content-type>
```

Note that the field's **`mime-type`** is set to **`application/json`**. This means that the field is configured to store JSON-formatted data (JSON is a popular data exchange format in web applications). For more information about this, see **section 4.1**.

This means that in order to import soft crop information with an image, you must include the appropriate field (as defined in your **`content-type`** resource) and insert the soft crop definitions, in JSON format in this field. For example:

```
<content source="ex" sourceid="20" type="picture" state="published">
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Croc</field>
  <field name="caption">A Croc on the beach</field>
  <field name="binary" title="crocodile">/tmp/escenic/import/croc.jpg</field>
  <field name="representations">
      {
```

```
        "thumbnail":
          {
            "crop":
              {
                "width":400,
                "height":400,
                "x":0,
                "y":54
              }
          },
        "narrow":
          {
            "crop":
              {
                "width":250,
                "height":200,
                "x":0,
                "y":54
              }
          },
        "wide":
          {
            "crop":
              {
                "width":400,
                "height":400,
                "x":102,
                "y":100
              }
          }
      }
  </field>
</content>
```

The important points to be aware of here are the following:

- The JSON structure **may** contain one object for each **`representation`** defined for the field in the **`content-type`** resource: you can, however omit some of the representations. If you omit an object for one of the **`representation`**s, then a default soft crop is created that has the aspect ratio defined in the **`content-type`** resource, is as large as possible and is positioned in the center of the image.
- A **`representation`** object's crop data must be encapsulated in a **`crop`** object.
- Each **`crop`** object must contain the following values:

  `x` **and** **`y`**
  : These coordinates define the crop start point in pixels, and are measured from the top-left corner of the image.

  `height` **and** **`width`**
  : These co-ordinates define the height and width (in pixels) of the area to be selected.

    If the **`representation`** definition in the **`content-type`** resource includes both an output height and width as in the examples above then the **`height`** and **`width`** values specified in the crop object should normally have a matching aspect ratio, as is the case in the **`thumbnail`** and **`narrow`** crop objects shown above. If you specify **`height`** and **`width`** values that do not match the **`representation`**'s aspect ratio, as is the case in the **`wide`** crop object shown above, then the Content Engine will crop the specified area but stretch or

compress it to fit the **`representation`**'s aspect ratio, resulting in a distorted image.

- White space is irrelevant in the JSON format: you can compress all of the **`representations`** field in the above example into a single line if you wish.
- Order is also irrelevant in JSON.

For detailed information about the JSON format, see http://www.json.org/.

4.2 Importing Content Items That Contain HTML Entities

The import service accepts only valid XML data. This means that the content in rich text fields must be valid XHTML, not HTML. This means that standard HTML named character entities such as **`&mdash;`** and **` `** are **not** allowed.

Here is an example of some content that includes such HTML entities.

```
<?xml version="1.0" encoding="UTF-8"?>
<escenic xmlns="http://xmlns.escenic.com/2009/import" version="2.0">
  <content source="ece-auto-gen" sourceid="6ecfd92e-12e3-4773-877c-0dff82811c29" ...>
    <uri>article68.ece</uri>
    ...
    <field name="body">
      <p xmlns="">Here are some HTML character entities: &mdash; and non-breaking space</p>
    </field>
    ...
  </content>
</escenic>
```

Two solutions to this problem are described below.

Adding a DOCTYPE declaration

Before importing, include a DOCTYPE declaration for the required characters. The example below shows the same import data with an in-line DOCTYPE declaration that defines the entities used in the content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE htmlEntities [
    <!ENTITY mdash "&#x2014;">
    <!ENTITY nbsp " ">
    ]>
<escenic xmlns="http://xmlns.escenic.com/2009/import" version="2.0">
  <content source="ece-auto-gen" sourceid="6ecfd92e-12e3-4773-877c-0dff82811c29" ...>
    <uri>article68.ece</uri>
    ...
    <field name="body">
      <p xmlns="">Here are some HTML character entities: &mdash; and non-breaking space</p>
    </field>
    ...
  </content>
</escenic>
```

Replacing the entities

Before importing, replace the named entities with valid numerical character entities. The example below shows the same import data processed in this way:

```
<?xml version="1.0" encoding="UTF-8"?>
<escenic xmlns="http://xmlns.escenic.com/2009/import" version="2.0">
  <content source="ece-auto-gen" sourceid="6ecfd92e-12e3-4773-877c-0dff82811c29" ...>
```

```
    <uri>article68.ece</uri>
    ...
    <field name="body">
      <p xmlns="">Here are some HTML character entities: &#x2014; and non-breaking space</p>
    </field>
    ...
  </content>
</escenic>
```

5 escenic-syndication

The **escenic-syndication** schema defines the **Escenic syndication format**. The Escenic syndication format is an open XML-based data format intended to simplify the exchange of content between the Escenic Content Engine and other systems. All content exported from Escenic Content Engine is exported to the Escenic syndication format, and the Content Engine can import any content that is presented in the form of a valid syndication format file. The syndication format is well-defined, so you can quite easily convert content between the syndication format and other XML-based formats using XSLT or similar tools.

You can use the syndication format import/export all types of content and metadata stored in an Escenic system. Note, however, that binary data is not included directly in syndication files: images, audio and video are imported from/exported to binary files, and are simply referenced from the syndication file.

Namespace URI

The namespace URI of the **escenic-syndication** schema is **http://xmlns.escenic.com/2009/import**.

Root Element

The root of an **escenic-syndication** file must be an **escenic** element.

5.1 area

Represents an area on an Escenic section page.

Syntax

```
<area
    name="text"
  >
    <content-ref>...</content-ref>*
    <list-ref>...</list-ref>*
    <group>...</group>*
  </area>
```

Child Elements

content-ref: **section 5.5**, **list-ref**: **section 5.15**, **group**: **section 5.12**.

Only one form of the **content-ref** element may be used: **Section Page content-ref** (**section 5.5.2**).

Examples

- This example shows an **area** element that contains a child **group**

```
<area name="center">
  <group name="twoCol">
```

```
    <area name="left">
      <list-ref source="ex" sourceid="s12" name="important" number-of-items="2"/>
    </area>
    <area name="right">
      <content-ref source="ex" sourceid="19">
        <field name="leadtext">New lead text</field>
      </content-ref>
      <content-ref source="ex" sourceid="20"/>
    </area>
  </group>
</area>
```

- This example shows an **area** element that contains **content-ref** elements.

```
<area name="right">
  <content-ref source="ex" sourceid="19">
    <field name="leadtext">New lead text</field>
  </content-ref>
  <content-ref source="ex" sourceid="20"/>
</area>
```

Attributes

name="*text*"
The name of this area. This must be the name of an area that is:

- Defined in the **layout-groups** resource of the target publication.
- Allowed in the current context.

"Allowed in the current context" means:

- If this **area** element is the child of a **section-page** element, then it must be the name of an area defined in the **layout-groups** resource as a child of the layout group specified with the **section-page** element's **layout-name** attribute.
- If this **area** element is the child of a **group** element, then it must be the name of an area defined in the **layout-groups** resource as a child of that group.

5.2 article-layout

The name of the article layout to use for the section.

Syntax

```
<article-layout>
    text
  </article-layout>
```

5.3 author

A reference to a content item author. Content item authors are themselves represented by **person** objects. A person object is a special type of content item containing the fields needed to hold the usual kinds of personal details (name, phone number, email address and so on).

An **author** element must therefore contain a reference to a person object in the target publication or a **person** element in the syndication file.

Syntax

```
<author
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    first-name="text"?
    last-name="text"?
    username="text"?
    email-address="text"?
    publication-name="text"?
  />
```

Examples

- This example imports a reference to an author (person). The referenced author must already exist in the database or appear before this element in the syndication file.

  ```
  <author username="m.cicero"/>
  ```

Attributes

`id-ref="`*`text`*`"` (optional)
The `id` of the author. If this attribute is specified, a `person` element with an `id` attribute that matches this attribute must appear somewhere **before** this `author` element in the syndication file.

If `dbid` or `source` and `sourceid` are specified, then this attribute is ignored.

`source="`*`text`*`"` (optional)
The `source` of the author. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `person` or `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `author` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`*`text`*`"` (optional)
The `sourceid` of the author. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `person` or `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `author` element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the author. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a **dbid** attribute that matches this attribute, **or**
- A **person** or **user** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **author** element in the syndication file.

first-name="*text*" (optional)
The first name of the this author. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a **first-name** field that matches this attribute, **or**
- A **person** or **user** element with a **field** called **first-name** that matches this attribute must appear somewhere **before** this **author** element in the syndication file.

Using the **first-name** attribute on its own is not recommended; you should use it in combination with the **last-name** attribute.

If **dbid** or **source** and **sourceid** or **id-ref** are specified, then this attribute is ignored.

last-name="*text*" (optional)
The surname of the this author. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a **last-name** field that matches this attribute, **or**
- A **person** or **user** element with a **field** called **last-name** that matches this attribute must appear somewhere **before** this **author** element in the syndication file.

Using the **last-name** attribute on its own is not recommended; you should use it in combination with the **first-name** attribute.

If **dbid** or **source** and **sourceid** or **id-ref** are specified, then this attribute is ignored.

username="*text*" (optional)
The username of the this author. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a **username** field that matches this attribute, **or**
- A **person** or **user** element with a **field** called **username** that matches this attribute must appear somewhere **before** this **author** element in the syndication file.

If **dbid** or **source** and **sourceid** or **id-ref** are specified, then this attribute is ignored.

email-address="*text*" (optional)
The email of the this author. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with an **email-address** field that matches this attribute, **or**
- A **person** or **user** element with a **field** called **email-address** that matches this attribute must appear somewhere **before** this **author** element in the syndication file.

If **dbid** or **source** and **sourceid** or **id-ref** are specified, then this attribute is ignored.

publication-name="*text*" (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **source** and **source-id** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

5.4 content

Represents an Escenic content item, which is a generic container for content: it could be a text article, an image or a multimedia object such as an audio or video object. The kind of content item represented by a **content** element is specified in the **type** attribute.

Syntax

```
<content
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    state="(draft|submitted|approved|published|deleted)"?
    type="text"?
    publishdate="text"?
    delete-relations="text"?
    activatedate="text"?
    expiredate="text"?
    creationdate="text"?
  >
    <section-ref/>*
    <relation>...</relation>*
    <reference/>*
    <field>...</field>*
    <tag/>*
    <update/>?
    <uri>...</uri>?
    <author/>*
    <creator/>?
    <priority/>?
  </content>
```

Child Elements

`section-ref`: **section 5.36**, **`relation`**: **section 5.22**, **`reference`**: **section 5.21**, **`field`**: **section 5.10**, **`tag`**: **section 5.37**, **`update`**: **section 5.39**, **`uri`**: **section 5.40**, **`author`**: **section 5.3**, **`creator`**: **section 5.6**, **`priority`**: **section 5.20**.

Only one form of the **`relation`** element may be used: **Typed relation** (**section 5.22.2**).

Only one form of the **`field`** element may be used: **Standard field** (**section 5.10.2**).

Examples

- This example imports a content item. It includes a reference to the author and to a section to which the content item is to belong. This section is set to be the content item's home section. Both the referenced author (i.e person) and the referenced section must already exist in the database or appear before this **`content`** element in the syndication file.

```
<content source="ex" sourceid="3" type="news" state="published">
  <author username="m.cicero"/>
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Ex Article 3</field>
  <field name="leadtext">Third article</field>
  <field name="body">
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc venenatis erat at nisl. In
 hac habitasse platea dictumst.</p>
    <p>Sed venenatis purus iaculis turpis.</p>
  </field>
</content>
```

- This example imports an "image" content item. It includes a reference to a section to which the content item is to belong. This section is set to be the content item's home section. The referenced section must already exist in the database or appear before this **`content`** element in the syndication file.

  Note the use of JSON syntax to include soft crop information in the **`representations`** field. For more information about this, see **section 4.1**.

```
<content source="ex" sourceid="20" type="picture" state="published">
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Croc</field>
  <field name="caption">A Croc on the beach</field>
  <field name="binary" title="crocodile">/tmp/escenic/import/croc.jpg</field>
  <field name="representations">
      {
        "thumbnail":
          {
            "crop":
              {
                "width":400,
                "height":400,
                "x":0,
                "y":54
              }
          },
        "narrow":
          {
            "crop":
              {
                "width":250,
                "height":200,
                "x":0,
                "y":54
```

```
                }
            },
          "wide":
            {
              "crop":
                {
                  "width":400,
                  "height":400,
                  "x":102,
                  "y":100
                }
            }
        }
    </field>
</content>
```

- This example imports a content item and inserts it into four different sections, two of which belong to a different publication. One of the local sections is set to be the content item's home section. In addition, one of the remote sections is set to be the content item's home section in that publication.

```
<content source="ex" sourceid="99" type="news" state="published">
  <section-ref unique-name="sports" home-section="true"/>
  <section-ref unique-name="ece_frontpage"/>
  <section-ref unique-name="sports" home-section="true" publication-name="other"/>
  <section-ref unique-name="ece_frontpage" publication-name="other"/>
  <field name="title">Local Cup Fiasco</field>
  <field name="leadtext">...</field>
  <field name="body">...</field>
</content>
```

Attributes

`id="`*`text`*`"` (optional)
A unique identifier for this **`content`** element. It is only valid and unique within the current syndication file and can be used to enable the establishment of relationships between elements in the file. Other elements in the file have **`id-ref`** attributes that can be used to reference **`content`** elements. If a **`content`** element does not have an **`id`** attribute then it must have either a **`dbid`** attribute or both a **`source`** and a **`sourceid`** attribute. A **`content`** element **may** have several or all of these attributes, in which case any of them can be used for establishing relationships.

The **`id`** attribute is not imported along with content items. Unless a **`dbid`** attribute has been specified, all imported content items are assigned new internal IDs during import.

`source="`*`text`*`"` (optional)
The name of the system from which this content item originates. Together with the **`sourceid`** attribute it forms a globally unique external identifier for the content item that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **`source`** and **`sourceid`** attributes that can be used for this purpose. If this attribute is specified then a **`sourceid`** attribute must also be specified. If a **`content`** element does not have a **`source`** and **`sourceid`** attribute then it must have either a **`dbid`** attribute or an **`id`** attribute. A **`content`** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If supplied, **source** and **sourceid** are imported and stored with content items. If **source** and **sourceid** are supplied and **dbid** is not supplied, then they are used to lookup an existing content item. If a content item with matching **source** and **sourceid** is found, then this content item is updated; otherwise a new content item is created.

If supplied, **source** and **sourceid** are imported and stored when creating new content items, but not when updating existing content items.

sourceid="*text*" (optional)
The id of this content item in the system from which it originates. Together with the **source** attribute it forms a globally unique external identifier for the **content** that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **source** and **sourceid** attributes that can be used for this purpose. If this attribute is specified then a **source** attribute must also be specified. If a **content** element does not have a **source** and **sourceid** attribute then it must have either a **dbid** attribute or an **id** attribute. A **content** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If **source** and **sourceid** are supplied and **dbid** is not supplied, then they are used to lookup an existing content item. If a content item with matching **source** and **sourceid** is found, then this content item is updated; otherwise a new content item is created.

If supplied, **source** and **sourceid** are imported and stored when creating new content items, but not when updating existing content items.

dbid="*text*" (optional)
The internal Content Engine ID of this content item, which can be used when importing updated versions of existing content items. It can also be used for establishing relationships between elements in the syndication file. Other elements in the file have **dbid** attributes that can be used for this purpose. If a **content** element does not have a **dbid** attribute then it must have either a **source** and **sourceid** attribute or an **id** attribute. A **content** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

You should only use the **dbid** attribute when importing updated versions of **existing** content items.

This attribute is never present in syndication files that have been exported from a database. The ID is always written to the **exported-dbid** attribute in exported syndication files.

---

exported-dbid="*text*" (optional)
The internal Content Engine ID of this content item, which can be used to identify the content item in the database from which it was exported.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

`state="(draft|submitted|approved|published|deleted)"` (optional)
The current state of this content item.

Allowed values are:

`draft` (default)
The content item is a draft. This is the default value when importing.

`submitted`
The content item is submitted for approval.

`approved`
The content item is approved for publishing.

`published`
The content item is published.

`deleted`
The content item has been deleted.

`type="`*`text`*`" (optional)`
Defines the type of content item represented by this **`content`** element. For import, the value specified must be the name of a content type as defined in the target publication's **`content-type`** resource. The value you specify here will then determine what kind of **`field`** elements the **`content`** element may own.

`publishdate="`*`text`*`" (optional)`
The date/time this content item was published, specified in the format:

```
yyyy-mm-dd hh:mm:ss.ffffffff
```

`delete-relations="`*`text`*`" (optional)`
If this attribute is set to **`true`** when re-importing an existing content item, then all the content item's existing relations are deleted.

`activatedate="`*`text`*`" (optional)`
The date/time this content item was/is to be activated, specified in the format:

```
yyyy-mm-dd hh:mm:ss.ffffffff
```

`expiredate="`*`text`*`" (optional)`
The date/time this content item expired/is to expire, specified in the format:

```
yyyy-mm-dd hh:mm:ss.ffffffff
```

`creationdate="`*text*`"` (optional)
The date/time this content item was created, specified in the format:

```
yyyy-mm-dd hh:mm:ss.ffffffff
```

If specified, this attribute is used when importing new content items that do not already exist in the database. It is, however, ignored when importing updates to content items that already exist. If it is omitted when importing a new content item, then the new content item's creation date is set to the current date.

5.5 content-ref

This element can appear in a number of different forms, described in the following sections.

5.5.1 List/Inbox content-ref

A reference to a content item.

Syntax

```
<content-ref
    id-ref="text"?
    publication-name="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
  />
```

Examples

- This example shows a list/inbox **`content-ref`** element. The **`source`** and **`source-id`** attributes identify the referenced content item. This content item must already exist in the database or appear before this element in the syndication file.

```
<content-ref source="ex" sourceid="13"/>
```

Attributes

`id-ref="`*text*`"` (optional)
The **`id`** of the referenced content item. If this attribute is specified, a **`content`** element with an **`id`** attribute that matches this attribute must appear somewhere **before** this **`content-ref`** element in the syndication file.

If **`dbid`** or **`source`** and **`sourceid`** are specified, then this attribute is ignored.

`publication-name="`*text*`"` (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **`source`** and **`source-id`** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the

referenced content item or section does not belong to the current publication.

`source="`*text*`" (optional)`
The `source` of the referenced content item. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `content` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `content-ref` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`*text*`" (optional)`
The `sourceid` of the referenced content item. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `content` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `content-ref` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`dbid="`*text*`" (optional)`
The `dbid` of the referenced content item. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a `dbid` attribute that matches this attribute, **or**
- A `content` element with a `dbid` attribute that matches this attribute must appear somewhere **before** this `content-ref` element in the syndication file.

This attribute is never present in syndication files that have been exported from a database. IDs are always written to `exported-dbid` attributes in exported syndication files.

`exported-dbid="`*text*`" (optional)`
The `dbid` of the referenced content item.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

5.5.2 Section Page content-ref

Represents a content-ref on an Escenic section page. The element attributes are used to identify the content item that the content-ref represents.

A **content-ref** element can contain child **field** elements in order to override the default contents of the summary fields displayed in the content-ref. The **name**s of any child **field**s must be valid summary fields for the content item referenced by the content-ref. Summaries are defined in the target publication's **content-type** resource.

Syntax

```
<content-ref
    id-ref="text"?
    publication-name="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
  >
    <field>...</field>*
    <options>...</options>?
  </content-ref>
```

Child Elements

field: section 5.10, **options**: section 5.17.

Only one form of the **field** element may be used: **Standard field** (section 5.10.2).

Examples

- This example shows a section-page **content-ref** element. The **source** and **source-id** attributes identify the referenced content item. This content item must already exist in the database or appear before this element in the syndication file. The optional **field** element inside the **content-ref** locally overrides the content of the same field in the referenced content item.

```
<content-ref source="ex" sourceid="19">
  <field name="leadtext">New lead text</field>
</content-ref>
```

Attributes

id-ref="*text*" (optional)
The **id** of the referenced content item. If this attribute is specified, a **content** element with an **id** attribute that matches this attribute must appear somewhere **before** this **content-ref** element in the syndication file. If the referenced content item does not belong to this content-ref's owning section, the it is automatically added during import.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

publication-name="*text*" (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **source** and **source-id** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

source="*text*" (optional)
The **source** of the referenced content item. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with **source** and **sourceid** attributes that match this element's **source** and **sourceid**, **or**
- A **content** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **content-ref** element in the syndication file.

If the referenced content item does not belong to this content-ref's owning section, the it is automatically added during import.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the referenced content item. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with **source** and **sourceid** attributes that match this element's **source** and **sourceid**, **or**
- A **content** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **content-ref** element in the syndication file.

If the referenced content item does not belong to this content-ref's owning section, the it is automatically added during import.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the referenced content item. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a **dbid** attribute that matches this attribute, **or**

- A **content** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **content-ref** element in the syndication file.

If the referenced content item does not belong to this content-ref's owning section, then it is automatically added during import.

This attribute is never present in syndication files that have been exported from a database. IDs are always written to **exported-dbid** attributes in exported syndication files.

---

exported-dbid="*text*" (optional)
The **dbid** of the referenced content item.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

5.6 creator

A reference to the creator of a content item. Content item authors are themselves represented by **person** objects. A person object is a special type of content item containing the fields needed to hold the usual kinds of personal details (name, phone number, email address and so on).

A **creator** element must therefore contain a reference to a person object in the publication or a **person** element in the syndication file.

Syntax

```
<creator
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    first-name="text"?
    last-name="text"?
    username="text"?
    email-address="text"?
    publication-name="text"?
  />
```

Attributes

id-ref="*text*" (optional)
The **id** of the creator. If this attribute is specified, a **person** element with an **id** attribute that matches this attribute must appear somewhere **before** this **creator** element in the syndication file.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

`source="`*text*`"` (optional)
The `source` of the creator. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `person` or `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `creator` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`*text*`"` (optional)
The `sourceid` of the creator. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `person` or `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `creator` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`dbid="`*text*`"` (optional)
The `dbid` of the creator. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a `dbid` attribute that matches this attribute, **or**
- A `person` or `user` element with a `dbid` attribute that matches this attribute must appear somewhere **before** this `creator` element in the syndication file.

`first-name="`*text*`"` (optional)
The first name of the this creator. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a `first-name` field that matches this attribute, **or**
- A `person` or `user` element with a `field` called `first-name` that matches this attribute must appear somewhere **before** this `creator` element in the syndication file.

Using the `first-name` attribute on its own is not recommended; you should use it in combination with the `last-name` attribute.

If `dbid` or `source` and `sourceid` or `id-ref` are specified, then this attribute is ignored.

`last-name="`*text*`"` (optional)
The surname of the this creator. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a `last-name` field that matches this attribute, **or**
- A `person` or `user` element with a `field` called `last-name` that matches this attribute must appear somewhere **before** this `creator` element in the syndication file.

Using the `last-name` attribute on its own is not recommended; you should use it in combination with the `first-name` attribute.

If `dbid` or `source` and `sourceid` or `id-ref` are specified, then this attribute is ignored.

`username="`*text*`"` (optional)
The username of the this creator. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a `username` field that matches this attribute, **or**
- A `person` or `user` element with a `field` called `username` that matches this attribute must appear somewhere **before** this `creator` element in the syndication file.

If `dbid` or `source` and `sourceid` or `id-ref` are specified, then this attribute is ignored.

`email-address="`*text*`"` (optional)
The email of the this creator. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with an `email-address` field that matches this attribute, **or**
- A `person` or `user` element with a `field` called `email-address` that matches this attribute must appear somewhere **before** this `creator` element in the syndication file.

If `dbid` or `source` and `sourceid` or `id-ref` are specified, then this attribute is ignored.

`publication-name="`*text*`"` (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the `source` and `source-id` attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

5.7 delete

Used to specify that the section referenced by this element's parent **section** is to be deleted from the target publication. The section can only be deleted if either:

- **recursive** is set to **true** or
- Both of the following conditions are satisfied:
  - It contains no child sections or **move-sections** is set to **true.**
  - It is not the home section of any content items or **delete-content** is set to **true**.

Syntax

```
<delete
    recursive="(true|false)"?
    delete-content="(true|false)"?
    move-sections="(true|false)"?
  />
```

Attributes

recursive="(true|false)" (optional)
If **true**, then all of the section's subsections and content items will be deleted with it.

delete-content="(true|false)" (optional)
If **true**, then all of the section's content items will be deleted with it.

move-sections="(true|false)" (optional)
If **true**, then all of the section's subsections will be moved to this section's parent section.

5.8 directory

A section "directory name". It is not really used as a directory name but is used as a component of the section URL.

Syntax

```
<directory>
    text
  </directory>
```

5.9 escenic

The root element of an Escenic syndication format file. The root element can contain any number of child elements, and the allowed types of child element can appear in any order, making the format very flexible. You can use it to import/export a complete publication in a single file, a single content item or anything in between.

Syntax

```
<escenic
    version="2.0"
  >
    <global-acl>...</global-acl>*
    <section-acl>...</section-acl>*
    <content>...</content>*
    <person>...</person>*
    <section>...</section>*
    <list>...</list>*
    <inbox>...</inbox>*
    <section-page>...</section-page>*
    <user>...</user>*
    <user-group>...</user-group>*
  </escenic>
```

Child Elements

global-acl: **section 5.11**, **section-acl**: **section 5.33**, **content**: **section 5.4**, **person**: **section 5.19**, **section**: **section 5.32**, **list**: **section 5.14**, **inbox**: **section 5.13**, **section-page**: **section 5.35**, **user**: **section 5.41**, **user-group**: **section 5.42**.

Attributes

version="2.0"

5.10 field

This element can appear in a number of different forms, described in the following sections.

5.10.1 Person field

Contains an item of personal information that will be added to the "person" content item.

Syntax

```
<field
    name="(description|first-name|middle-name|last-name|occupation|address|email-address|phone-work-direct|phone-mobile|phone-private)"
  >
    text
  </field>
```

Attributes

name="(description|first-name|middle-name|last-name|occupation|address|email-address|phone-work-direct|phone-mobile|phone-private)"

Indicates the type of information in this field.

Allowed values are:

description

first-name

middle-name

last-name

occupation

address

email-address

phone-work-direct

phone-mobile

phone-private

5.10.2 Standard field

Represents one field in a content item or relation. The element's content model **appears** to allow almost anything, but in practice this is not the case. When importing, the **field** element content is expected to conform to a field definition identified by the **name** attribute, and will fail to be imported if this is not the case.

"Link" fields

If the field definition identified by the **name** attribute has the **type "link"**, then the field is expected to contain the path of a binary file that is to be uploaded (for example, an image or other multimedia file, a PDF file, word processing document or spreadsheet). The referenced object must be located somewhere on the server.

If you are using Web Studio to import the syndication file, then the field must contain the absolute path of the binary file, for example:

```
<field name="image" title="Def">/tmp/import/def.jpg</field>
```

If you are using the import service, however, then the field can contain either an absolute or a relative path (relative to the **importDirectory** as defined in the import task configuration file):

```
<field name="image" title="Ghi">ghi.jpg</field>
```

For information about import task configuration files, see **section 2.2.1.1**.

In-line relations

If the field definition identified by the **name** attribute has the **type "basic"** and the **mime-type "application/xhmtl+xml"**, then the field may contain in-line relations to other content items such as images or related articles. These in-line relations are represented by **relation** elements included in the XHTML/XML markup in the field.

Schedule fields

If the field definition identified by the **name** attribute has the **type "schedule"**, then the field must contain a **schedule:schedule** element (**section 5.30**) containing the value of the schedule field.

Arrays

If the field definition identified by the **name** attribute specifies that this is an array, then the field must contain a sequence of 0 or more **value** elements containing the members of the array.

Complex fields

If the field definition identified by the **name** attribute has the **type "complex"**, then the field must contain a sequence of 0 or more **field** elements containing the members of complex field.

Syntax

```
<field
    name="text"
    title="text"?
  >
    ((ANY-FOREIGN-ELEMENT|<relation>...</relation>|text)*|<field>...</field>*|<value>...</
value>*|<schedule:schedule>...</schedule:schedule>)
    <options>...</options>?
  </field>
```

Child Elements

relation: section 5.22, text, **field**: section 5.10, **value**: section 5.45, **schedule:schedule**: section 5.30, **options**: section 5.17.

Only one form of the **field** element may be used: **Standard field** (section 5.10.2).

Attributes

name="*text*"
Identifies the content item/relation field represented by this **field** element. For import, the value specified must be the name of one of the fields defined for the content item/relation in the target publication's **content-type** resource. The value you specify here will then determine what kind of content your **field** element may have.

If, for example, your **field** element has the **name "headline"** and belongs to a **content** element with the **type "news"**, then:

- The **content-type** resource of the target publication must contain a **content-type** element with the name **"news"**.
- The **"news" content-type** element must contain a **field** element with the name **"headline"**.
- The content of your **field** element must conform to the **"headline"** field definition in the **content-type** resource.

title="*text*" (optional)
A title associated with the field. This attribute is only used for **link** fields (fields that are defined in the **content-type** resource as having the **type "link"**). It is used to hold an alternative name for the resource referenced in the field. It could be used, for example, to contain a descriptive title (e.g **"London Bridge"**) for a link field containing the URL of an image file with a cryptic auto-generated name (e.g **image-store://places/img099345.jpg**).

5.10.3 User field

Contains an item of personal information that will be added to or replaced in the "user" content item.

Syntax

```
<field
    name="(description|first-name|middle-name|last-name|occupation|address|email-address|phone-work-
direct|phone-mobile|phone-private|username|password)"
  >
    text
  </field>
```

Attributes

name="(description|first-name|middle-name|last-name|occupation|address|email-address|phone-work-direct|phone-mobile|phone-private|username|password)"

Indicates the type of information in this field. Note that you cannot modify the **username** of an existing user.

Allowed values are:

description

first-name

middle-name

last-name

occupation

address

email-address

phone-work-direct

phone-mobile

phone-private

username

password

5.11 global-acl

Represents an Escenic **access control list (ACL)**, which assigns a specified **role** to one or more users or user groups. A role implies a defined set of access rights. This element represents a **global** ACL, so the access rights apply to the publication as a whole.

Syntax

```
<global-acl
    name="(reader|administrator|useradmin|editor|journalist)"
    publication-id="integer"
  >
    <user-ref/>*
    <user-group-ref/>*
  </global-acl>
```

Attributes

name="(reader|administrator|useradmin|editor|journalist)"
The name of the **role** represented by this ACL.

Allowed values are:

reader

administrator

useradmin

editor

journalist

publication-id="*integer*"
The ID of the publication to which this **global-acl** belongs.

5.12 group

Represents a layout group (a group of areas) on an Escenic section page.

Syntax

```
<group
    name="text"
  >
    <options>...</options>?
    <area>...</area>+
  </group>
```

Examples

- This example shows a **group** element that defines a two-column layout.

```
<group name="twoCol">
  <area name="left">
    <list-ref source="ex" sourceid="s12" name="important" number-of-items="2"/>
  </area>
  <area name="right">
    <content-ref source="ex" sourceid="19">
      <field name="leadtext">New lead text</field>
    </content-ref>
    <content-ref source="ex" sourceid="20"/>
  </area>
</group>
```

Attributes

name="*text*"
The name of this layout group. This must be the name of a group that is:

- Defined in the **layout-groups** resource of the target publication.
- Allowed in the current context.

"Allowed in the current context" means that it must be one of the groups defined in the **layout-groups** resource as a child of the area with the same name as this element's parent **area** element.

5.13 inbox

Represents an **inbox**. An inbox is a list of content items. Inboxes belong to sections and are generally used by section editors to organize publication work flow. A content item may not belong to more than one inbox at a time The content items in an inbox are represented by the **inbox** element's child **content-ref** elements.

Syntax

```
<inbox
    name="text"?
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    unique-name="text"?
    action="(remove|insert)"?
  >
    <content-ref/>*
  </inbox>
```

Examples

- This example imports an inbox called "review". The **source** and **source-id** attributes identify the section to which the inbox is to be added. This section must already exist in the database or appear before this element in the syndication file, as must the content item that is included in the inbox.

```
<inbox source="ex" sourceid="s12" name="review">
  <content-ref source="ex" sourceid="13"/>
</inbox>
```

Attributes

name="*text*" (optional)
The name of this inbox. To import content items to the default inbox called "Inbox", either omit this attribute or specify **name=""**. Do **not** specify **name="Inbox"**, as this will create a second inbox with the name "Inbox".

id-ref="*text*" (optional)
The **id** of the section to which this inbox is to be added. If this attribute is specified, a **section** element with an **id** attribute that matches this attribute must appear somewhere **before** this **inbox** element in the syndication file.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the section to which this inbox is to be added. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**

- A `section` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `inbox` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`***text***`"` **(optional)**
The `sourceid` of the section to which this inbox is to be added. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with `source` and `sourceid` attributes that match `source` and `sourceid`, **or**
- A `section` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `inbox` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`dbid="`***text***`"` **(optional)**
The `dbid` of the section to which this inbox is to be added. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a `dbid` attribute that matches this attribute, **or**
- A `section` element with a `dbid` attribute that matches this attribute must appear somewhere **before** this `inbox` element in the syndication file.

`exported-dbid="`***text***`"` **(optional)**

`unique-name="`***text***`"` **(optional)**
The `unique-name` or `name` of the section to which this inbox is to be added. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a `uniquename` or `name` attribute that matches this attribute, **or**
- A `section` element with a `unique-name` or `name` attribute that matches this attribute must appear somewhere **before** this `inbox` element in the syndication file.

If this is not the case, or if there is a matching `name` attribute but it is not unique, then import will fail.

If `dbid` or `source` and `sourceid` or `id` are specified, then this attribute is ignored.

`action="(`**remove**`|`**insert**`)"` **(optional)**
Determines what action is taken during import if the section page already exists.

Allowed values are:

`remove`
The inbox is cleared before import.

`insert` (default)
The inbox is not cleared before import: new entries are simply appended to the inbox.

5.14 list

Represents an Escenic **list**. An Escenic list is an ordered list of content items that are related in some way (a list of articles related to an ongoing news issue, for example). The content items in a list are represented by the **`list`** element's child **`content-ref`** elements.

Syntax

```
<list
    name="text"
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    unique-name="text"?
    action="(remove|insert)"?
  >
    <content-ref/>*
  </list>
```

Examples

- This example imports a list called "important". The **`source`** and **`source-id`** attributes identify the section to which the list is to be added. This section must already exist in the database or appear before this element in the syndication file, as must the content items that are included in the list.

```
<list source="ex" sourceid="s12" name="important">
  <content-ref source="ex" sourceid="13"/>
</list>
```

Attributes

`name="`*`text`*`"`
The name of this list.

`id-ref="`*`text`*`"` (optional)
The **`id`** of the section to which this list is to be added. If this attribute is specified, a **`section`** element with an **`id`** attribute that matches this attribute must appear somewhere **before** this **`list`** element in the syndication file.

If **`dbid`** or **`source`** and **`sourceid`** are specified, then this attribute is ignored.

`source="`*`text`*`"` (optional)
The **`source`** of the section to which this list is to be added. If this attribute is specified, then **`sourceid`** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **list** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the section to which this list is to be added. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **list** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the section to which this list is to be added. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a **dbid** attribute that matches this attribute, **or**
- A **section** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **list** element in the syndication file.

exported-dbid="*text*" (optional)

unique-name="*text*" (optional)
The **unique-name** or **name** of the section to which this list is to be added. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a **uniquename** or **name** attribute that matches this attribute, **or**
- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **list** element in the syndication file.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

action="(remove|insert)" (optional)
Determines what action is taken during import if the section page already exists.

Allowed values are:

`remove`
The list is cleared before import.

`insert` (default)
The list is not cleared before import: new entries are simply appended to the list.

5.15 list-ref

A reference to a list that appears in a section page area.

Syntax

```
<list-ref
    name="text"
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    unique-name="text"?
    number-of-items="int"?
    publication-name="text"?
  >
    <options>...</options>?
  </list-ref>
```

Examples

- This example shows a **`list-ref`** element used to place the first two content items from the list "important" in an area. The **`source`** and **`source-id`** attributes identify the section to which the referenced list belongs. Both this section and the list must already exist in the database or appear before this element in the syndication file

```
<area name="left">
  <list-ref source="ex" sourceid="s12" name="important" number-of-items="2"/>
</area>
```

Attributes

`name="`*`text`*`"`
The name of the list to which this list-ref refers. A list is not uniquely referenced by its name. You must therefore also identify the section to which the list belongs by specifying one of:

- **`id-ref`**
- **`source`** and **`sourceid`**
- **`dbid`**
- **`unique-name`**

`id-ref="`*`text`*`"` (optional)
The **`id`** of the section to which this list-ref is to be added. If this attribute is specified, a **`section`** element with an **`id`** attribute that matches this attribute must appear somewhere **before** this **`list-ref`** element in the syndication file.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the section to which this list-ref is to be added. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **list-ref** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the section to which this list-ref is to be added. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **list-ref** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the section to which this list-ref is to be added. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a **dbid** attribute that matches this attribute, **or**
- A **section** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **list-ref** element in the syndication file.

exported-dbid="*text*" (optional)

unique-name="*text*" (optional)
The **unique-name** or **name** of the section to which this list-ref is to be added. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a **uniquename** or **name** attribute that matches this attribute, **or**
- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **list-ref** element in the syndication file.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

number-of-items="*int*" (optional)
The number of items from this list that are to appear in an area. The specified number of items are taken from the top of the list.

publication-name="*text*" (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **source** and **source-id** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

5.16 mirror-source

A reference to a section that this element's owning section is to mirror. The owning section has no content of its own, but just mirrors the content of the section referenced here. The owning section's **mirror-source** attribute may **not** be set to **true**.

Syntax

```
<mirror-source
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    (unique-name="text")?
    publication-name="text"?
  />
```

Examples

- This example imports a mirror section and illustrates the use of the **mirror-source** element to reference the section that is to be mirrored. The referenced section must already exist in the database or appear before this element in the syndication file.

```
<section source="ex" sourceid="s5" name="example-mirror-target" mirror-source="true">
  <parent unique-name="ece_incoming"/>
  <mirror-source unique-name="example-mirror-source"/>
</section>
```

Attributes

id-ref="*text*" (optional)
The **id** of the section this section is to mirror. If this attribute is specified, a **section** element with an **id** attribute that matches this attribute must appear somewhere **before** this **mirror-source** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the section this section is to mirror. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **mirror-source** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the section this section is to mirror. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **mirror-source** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the section this section is to mirror. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with a **dbid** attribute that matches this attribute, **or**
- A **section** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **mirror-source** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

exported-dbid="*text*" (optional)

unique-name="*text*" (optional)
The **unique-name** or **name** of the section this section is to mirror. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a mirrorable section with a **uniquename** or **name** attribute that matches this attribute, **or**
- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **mirror-**

source element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

publication-name="*text*" (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **source** and **source-id** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

5.17 options

Represents a set of options (name-value pairs) stored in fields.

Syntax

```
<options>
    <field>...</field>*
  </options>
```

Child Elements

field: **section 5.10**.

Only one form of the **field** element may be used: **Standard field** (**section 5.10.2**).

5.18 parent

A reference to this section's parent section. This element makes it possible to establish section tree relationships during import. Note that the Content Engine's import subsystem does not tolerate unusually deep section hierarchies. Section trees that are more than 40 sections deep cannot be guaranteed to import successfully.

Syntax

```
<parent
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    unique-name="text"?
    inherit-access-control-list="(true|false)"?
  />
```

Examples

- This example imports a section. The child **parent** element specifies where the section is to be inserted into the section hierarchy. The referenced section must already exist in the database or appear before this element in the syndication file.

```
<section source="ex" sourceid="s2" name="example-section">
  <parent unique-name="ece_incoming"/>
</section>
```

Attributes

id-ref="*text*" (optional)
The **id** of the parent section. If this attribute is specified, a **section** element with an **id** attribute that matches this attribute must appear somewhere **before** this **parent** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the parent section. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **parent** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the parent section. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **parent** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the parent section. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a mirrorable section with a **dbid** attribute that matches this attribute, **or**
- A **section** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **parent** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

This attribute is never present in syndication files that have been exported from a database. IDs are always written to **exported-dbid** attributes in exported syndication files.

---

exported-dbid="*text*" (optional)
The **dbid** of the parent section.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

---

unique-name="*text*" (optional)
The **unique-name** or **name** of the parent section. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a mirrorable section with a **uniquename** or **name** attribute that matches this attribute, **or**
- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **parent** element in the syndication file. The referenced **section** element must have its **mirror-source** attribute set to **true**.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

inherit-access-control-list="(true|false)" (optional)
If set to **false** then the section will not inherit the access control list of the parent section, and the section will maintain its own access control list.

Most sections should inherit the access control list of the parent. In a large section tree it is recommended that only a few of the sections have their own access control lists, since a large number of access control lists may impede search performance.

5.19 person

Represents a special kind of content item used to store personal information about persons related to a publication (mostly contributors and editors of various kinds).

Syntax

```
<person
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
  >
    <field>...</field>*
  </person>
```

Child Elements

`field`: **section 5.10**.

Only one form of the **`field`** element may be used: **Person field** (**section 5.10.1**).

Examples

- This example imports a publication user (a person with user name and password for accessing the publication).

```
<user source="ex" sourceid="21">
  <field name="first-name">Marcus</field>
  <field name="last-name">Cicero</field>
  <field name="email-address">marcus@cicero.org</field>
  <field name="username">m.cicero</field>
  <field name="password">tullius</field>
</user>
```

Attributes

`id="`*text*`"` (optional)
A unique identifier for this **`person`** element. It is only valid and unique within the current syndication file and can be used to enable the establishment of relationships between elements in the file. Other elements in the file have **`id-ref`** attributes that can be used to reference **`person`** elements. If a **`person`** element does not have an **`id`** attribute then it must have either a **`dbid`** attribute or both a **`source`** and a **`sourceid`** attribute. A **`person`** element **may** have several or all of these attributes, in which case any of them can be used for establishing relationships.

The **`id`** attribute is not imported along with persons. Unless a **`dbid`** attribute has been specified, all imported persons are assigned new internal IDs during import.

`source="`*text*`"` (optional)
The name of the system from which this person originates. Together with the **`sourceid`** attribute it forms a globally unique external identifier for the person that can be used for establishing relationships between

elements in the syndication file. Other elements in the file have **source** and **sourceid** attributes that can be used for this purpose. If this attribute is specified then a **sourceid** attribute must also be specified. If a **person** element does not have a **source** and **sourceid** attribute then it must have either a **dbid** attribute or an **id** attribute. A **person** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If supplied, **source** and **sourceid** are imported and stored with persons. If **source** and **sourceid** are supplied and **dbid** is not supplied, then they are used to lookup an existing person. If a person with matching **source** and **sourceid** is found, then this person is updated; otherwise a new person is created.

If supplied, **source** and **sourceid** are imported and stored when creating new persons, but not when updating existing persons.

sourceid="*text*" (optional)
The id of this person in the system from which it originates. Together with the **source** attribute it forms a globally unique external identifier for the **person** that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **source** and **sourceid** attributes that can be used for this purpose. If this attribute is specified then a **source** attribute must also be specified. If a **person** element does not have a **source** and **sourceid** attribute then it must have either a **dbid** attribute or an **id** attribute. A **person** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If **source** and **sourceid** are supplied and **dbid** is not supplied, then they are used to lookup an existing person. If a person with matching **source** and **sourceid** is found, then this person is updated; otherwise a new person is created.

If supplied, **source** and **sourceid** are imported and stored when creating new persons, but not when updating existing persons.

dbid="*text*" (optional)
The internal Content Engine ID of this person, which can be used when importing updated versions of existing content items. It can also be used for establishing relationships between elements in the syndication file. Other elements in the file have **dbid** attributes that can be used for this purpose. If a **person** element does not have a **dbid** attribute then it must have either a **source** and **sourceid** attribute or an **id** attribute. A **person** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

You should only use the **dbid** attribute when importing updated versions of **existing** persons.

This attribute is never present in syndication files that have been exported from a database. The ID is always written to the **exported-dbid** attribute in exported syndication files.

---

exported-dbid="*text*" (optional)
The internal Content Engine ID of this person, which can be used to identify the person in the database from which it was exported.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

5.20 priority

Used to set section priority.

Syntax

```
<priority
    value="int"?
  />
```

Attributes

value="*int*" (optional)
Section priority.

5.21 reference

Use of this element is deprecated. It is only retained for reasons of backwards compatibility.

Syntax

```
<reference
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    type="(person|link.relatedArticle|link.relatedSite|image|media|profile)"?
    duplicates="(true|false)"?
    element="text"?
    align="text"?
    alttext="text"?
    caption="text"?
    version="text"?
  />
```

5.22 relation

This element can appear in a number of different forms, described in the following sections.

5.22.1 In-line relation

Represents a relationship between the content item represented by this element's owning **content** element, and another content item. This form of the **relation** element may appear in-line in **field**s with XHTML content.

Syntax

```
<relation
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    publication-name="text"?
  >
    <field>...</field>*
  </relation>
```

Child Elements

field: **section 5.10**.

Only one form of the **field** element may be used: **Standard field** (**section 5.10.2**).

Examples

- This example shows an "in-line" **relation** element used to include a link to another content item in a field. The field must be defined with **mime-type="application/xhmtl+xml"** in the **content-type** resource. Note that when used in-line, the **relation** element has no **type** attribute.

```
<field name="body">
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc venenatis erat at nisl. In hac
 habitasse platea dictumst.</p>
  <p>
    <relation source="ex" sourceid="13"/>
  </p>
</field>
```

- This example shows a **relation** used in-line to include an "image" content item.

```
<content source="ex" sourceid="17" type="news" state="published">
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Ex Article 7</field>
  <field name="leadtext">Seventh article</field>
  <field name="body">
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc venenatis erat at nisl. In
 hac habitasse platea dictumst.</p>
    <p>
      <relation source="ex" sourceid="20"/>
    </p>
  </field>
</content>
```

Attributes

id-ref="*text*" (optional)
The **id** of the related content item. If this attribute is specified, a **content** element with an **id** attribute that matches this attribute must appear somewhere **before** this **relation** element in the syndication file.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the related content item. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with **source** and **sourceid** attributes that match this element's **source** and **sourceid**, **or**
- A **content** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **relation** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

sourceid="*text*" (optional)
The **sourceid** of the related content item. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with **source** and **sourceid** attributes that match this element's **source** and **sourceid**, **or**
- A **content** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **relation** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

dbid="*text*" (optional)
The **dbid** of the related content item. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a content item with a **dbid** attribute that matches this attribute, **or**
- A **content** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **relation** element in the syndication file.

This attribute is never present in syndication files that have been exported from a database. IDs are always written to **exported-dbid** attributes in exported syndication files.

---

exported-dbid="*text*" (optional)
The **dbid** of the related content item.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

`publication-name="`*`text`*`"` (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **`source`** and **`source-id`** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

5.22.2 Typed relation

Represents a relationship between the content item represented by this element's owning **`content`** element, and another content item. This form of the **`relation`** element may only appear as a direct child of a **`content`** element. **`relation`** elements that appear in-line inside **`field elements`** may not have a **`type`** attribute.

Syntax

```
<relation
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    publication-name="text"?
    type="text"
  >
    <field>...</field>*
  </relation>
```

Child Elements

`field`: **section 5.10**.

Only one form of the **`field`** element may be used: **Standard field** (**section 5.10.2**).

Examples

- This example shows a **`relation`** element used to insert a relation to an "image" content item. The optional **`field`** element inside the **`relation`** locally overrides the content of the same field in the referenced content item.

```
<content source="ex" sourceid="13" type="news" state="published">
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Ex Article 5</field>
  <field name="leadtext">Fifth article</field>
  <field name="body">
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc venenatis erat at nisl. In
 hac habitasse platea dictumst.</p>
    <p>Sed venenatis purus iaculis turpis.</p>
  </field>
  <relation source="ex" sourceid="20" type="slides">
    <field name="title">Croc on beach</field>
  </relation>
```

```
</content>
```

Attributes

`id-ref="`*`text`*`"` (optional)
The `id` of the related content item. If this attribute is specified, a `content` element with an `id` attribute that matches this attribute must appear somewhere **before** this `relation` element in the syndication file.

If `dbid` or `source` and `sourceid` are specified, then this attribute is ignored.

`source="`*`text`*`"` (optional)
The `source` of the related content item. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `content` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `relation` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`*`text`*`"` (optional)
The `sourceid` of the related content item. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a content item with `source` and `sourceid` attributes that match this element's `source` and `sourceid`, **or**
- A `content` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `relation` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`dbid="`*`text`*`"` (optional)
The `dbid` of the related content item. If this attribute is specified, then one of the following two conditions must be satisfied:

- The target publication must already contain a content item with a `dbid` attribute that matches this attribute, **or**
- A `content` element with a `dbid` attribute that matches this attribute must appear somewhere **before** this `relation` element in the syndication file.

This attribute is never present in syndication files that have been exported from a database. IDs are always written to **`exported-dbid`** attributes in exported syndication files.

`exported-dbid="`*text*`"` (optional)
The **`dbid`** of the related content item.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

`publication-name="`*text*`"` (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **`source`** and **`source-id`** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

`type="`*text*`"`
Defines the type of relation represented by this **`relation`** element. For import, the value specified must be the name of a relation type as defined in the target publication's **`content-type`** resource. The value you specify here will determine how the relationship defined by this element is presented both in the publication and in Content Studio.

5.23 schedule:daily

A **`daily`** element defines the schedule of an event that recurs daily.

This element belongs to the **`http://xmlns.escenic.com/2011/schedule`** namespace. The conventional prefix for this namespace is **`schedule`**.

Syntax

```
<schedule:daily
    start-time="text"
    end-time="text"
  />
```

Attributes

`start-time="`*text*`"`
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

`end-time="`*text*`"`
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

5.24 schedule:exception

This element can appear in a number of different forms, described in the following sections.

5.24.1 Recurring schedule:exception

This form of the **exception** element represents an exception to a schedule that recurs on a weekly or monthly basis. An exception to a schedule is a day on which the scheduled event either:

- Does not occur at all if **start-time** and **end-time** are omitted, or
- Occurs at a different time than specified in the general schedule, as specified by **start-time** and **end-time**

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:exception>
    (<schedule:weekly>...</schedule:weekly>|<schedule:monthly/>)
  </schedule:exception>
```

Child Elements

schedule:weekly: **section 5.31**, **schedule:monthly**: **section 5.25**.

Only one form of the **schedule:weekly** element may be used: **Exception weekly** (**section 5.31.1**).

Only one form of the **schedule:monthly** element may be used: **Exception monthly** (**section 5.25.1**).

5.24.2 Single schedule:exception

This form of the **exception** element represents an exception to a schedule that occurs only once. An exception to a schedule is a day on which the scheduled event either:

- Does not occur at all if **start-time** and **end-time** are omitted, or
- Occurs at a different time than specified in the general schedule, as specified by **start-time** and **end-time**

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:exception
    date="text"
    start-time="text" end-time="text"?
```

```
/>
```

Attributes

`date="`*`text`*`"`
The date on which the exception occurs. The date must be specified in ISO-8601 format - that is, *YYYY-MM-DD*.

`start-time="`*`text`*`"` (optional)
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

`end-time="`*`text`*`"` (optional)
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

5.25 schedule:monthly

This element can appear in a number of different forms, described in the following sections.

5.25.1 Exception schedule:monthly

This form of the **`monthly`** element defines schedule exceptions that recur every month. It can only be used to define one exception per month.

On the specified day, the scheduled event either:

- Does not occur at all if **`start-time`** and **`end-time`** are omitted, or
- Occurs at a different time than specified in the general schedule, as specified by **`start-time`** and **`end-time`**

This element belongs to the **`http://xmlns.escenic.com/2011/schedule`** namespace. The conventional prefix for this namespace is **`schedule`**.

Syntax

```
<schedule:monthly
    start-time="text" end-time="text"?
    ( date="positiveInteger" | week-of-month="positiveInteger" day="text" )
  />
```

Attributes

`start-time="`*`text`*`"` (optional)
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

`end-time="`*`text`*`"` (optional)
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

`date="`*`positiveInteger`*`"`
The day of the month on which the exception occurs, specified as an integer between 1 and 31.

week-of-month="*positiveInteger*"
The week of the month in which the exception occurs, specified as an integer between 1 and 5.

day="*text*"
The day of the week on which the exception occurs.

One of the following English weekday names must be used:

monday
tuesday
wednesday
thursday
friday
saturday
sunday

Abbreviations are not allowed, and the interpreter is case-sensitive (upper case/mixed case values will **not** be accepted).

5.25.2 Schedule schedule:monthly

This form of the **monthly** element defines the schedule of an event that recurs at monthly intervals. The event may occur only once each month.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:monthly
    start-time="text"
    end-time="text"
    ( date="positiveInteger" | week-of-month="positiveInteger" day="text" )
  />
```

Attributes

start-time="*text*"
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

end-time="*text*"
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

date="*positiveInteger*"
The day of the month on which the event occurs, specified as an integer between 1 and 31.

week-of-month="*positiveInteger*"
The week of the month in which the event occurs, specified as an integer between 1 and 5.

day="*text*"
The day of the week on which the event occurs.

One of the following English weekday names must be used:

monday
tuesday
wednesday
thursday
friday
saturday
sunday

Abbreviations are not allowed, and the interpreter is case-sensitive (upper case/mixed case values will **not** be accepted).

5.26 schedule:occurrence

An **occurrence** element defines the schedule of an event that occurs only once.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:occurrence
    date="text"
    start-time="text"
    end-time="text"
  />
```

Attributes

date="*text*"
The date on which the event occurs. The date must be specified in ISO-8601 format - that is, *YYYY-MM-DD*.

start-time="*text*"
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

end-time="*text*"
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

5.27 schedule:pattern

This element can appear in a number of different forms, described in the following sections.

5.27.1 Exception schedule:pattern

This form of the **pattern** element defines the characteristics of a weekly recurring exception. On the day specified by the **weekdays** attribute, the scheduled event either:

- Does not occur at all if **start-time** and **end-time** are omitted, or
- Occurs at a different time than specified in the general schedule, as specified by **start-time** and **end-time**

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:pattern
    weekdays="text"
    start-time="text" end-time="text"?
  />
```

Attributes

weekdays="*text*"
The weekday on which the exception occurs. Note that only one weekday can be specified in this attribute. For example:

```
weekdays="monday"
```

The following English weekday names must be used:

monday
tuesday
wednesday
thursday
friday
saturday
sunday

Abbreviations are not allowed, and the interpreter is case-sensitive (upper case/mixed case values will **not** be accepted).

start-time="*text*" (optional)
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

end-time="*text*" (optional)
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

5.27.2 Schedule schedule:pattern

This form of the **pattern** element defines the characteristics of a weekly recurring event.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:pattern
    weekdays="text"
    start-time="text"
```

```
    end-time="text"
  />
```

Attributes

weekdays="*text*"
The weekdays on which the event occurs, specified in a comma-separated list. For example:

```
weekdays="monday,wednesday,friday"
```

The following English weekday names must be used:

monday
tuesday
wednesday
thursday
friday
saturday
sunday

Abbreviations are not allowed, and the interpreter is case-sensitive (upper case/mixed case values will **not** be accepted).

start-time="*text*"
The event start time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

end-time="*text*"
The event end time, specified in ISO.8601 format - that is, *hh*:*mm*:*ss*.

5.28 schedule:range

A **range** element defines the period during which a recurring event is to recur.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:range
    start-date="text"
    end-date="text"
  />
```

Attributes

start-date="*text*"
The date on which the period defined by this **range** element begins. The date must be specified in ISO-8601 format - that is, *YYYY-MM-DD*.

end-date="*text*"
The date on which the period defined by this **range** element ends. The date specified here is included in the period. The date must be specified in ISO-8601 format - that is, *YYYY-MM-DD*.

5.29 schedule:recurrence

A **recurrence** element defines the schedule of an event that occurs more than once.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:recurrence>
    (<schedule:daily/>|<schedule:weekly>...</schedule:weekly>|<schedule:monthly/>)
    <schedule:range/>
    <schedule:exception>...</schedule:exception>*
  </schedule:recurrence>
```

Child Elements

schedule:daily: section 5.23, **schedule:weekly**: section 5.31, **schedule:monthly**: section 5.25, **schedule:range**: section 5.28, **schedule:exception**: section 5.24.

Only one form of the **schedule:weekly** element may be used: **Schedule weekly** (section 5.31.2).

Only one form of the **schedule:monthly** element may be used: **Schedule monthly** (section 5.25.2).

The following forms of the **schedule:exception** element may be used: **Single exception** (section 5.24.2), **Recurring exception** (section 5.24.1).

5.30 schedule:schedule

Represents a **schedule**. A schedule is a set of elements that represents a series of one or more dates or date/time instances. A schedule is intended to enable the definition of times for both single and recurring events in a standardized and straightforward manner. It can be used to define meeting schedules, opening hours and so on It allows for a range of different recurrence intervals, the definition of exceptions ("closed on Mondays") and so on.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:schedule
    time-zone="text"
  >
    (<schedule:occurrence/>|<schedule:recurrence>...</schedule:recurrence>)+
  </schedule:schedule>
```

Attributes

time-zone="*text*"
The primary time zone of this **schedule**. It specifies the time zone in which the event defined by the **schedule** occurs. The time zone must be specified using a standard **TZ** time zone database name such as **Europe/Oslo**. You can find a complete list of all valid TZ time zone names at http://en.wikipedia.org/wiki/List_of_tz_database_time_zones.

5.31 schedule:weekly

This element can appear in a number of different forms, described in the following sections.

5.31.1 Exception schedule:weekly

This form of the **weekly** element defines schedule exceptions that recur every week. There may be several exceptions each week, as defined by the pattern element.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:weekly>
    <schedule:pattern/>
  </schedule:weekly>
```

Child Elements

schedule:pattern: **section 5.27**.

Only one form of the **schedule:pattern** element may be used: **Exception pattern** (**section 5.27.1**).

5.31.2 Schedule schedule:weekly

This form of the **weekly** element defines the schedule of an event that recurs at intervals of one or more weeks. The event may occur several times each week, as defined by the **pattern** element.

This element belongs to the **http://xmlns.escenic.com/2011/schedule** namespace. The conventional prefix for this namespace is **schedule**.

Syntax

```
<schedule:weekly
    interval="positiveInteger"?
  >
    <schedule:pattern/>+
  </schedule:weekly>
```

Child Elements

schedule:pattern: section 5.27.

Only one form of the **schedule:pattern** element may be used: **Schedule pattern** (section 5.27.2).

Attributes

interval="*positiveInteger*" (optional)
The number of weeks between each occurrence of the event. The value specified must be 1 or greater. The default is 1.

5.32 section

Represents a section of an Escenic publication. Section tree relationships are created using this element's child **parent** element, which references the section to which this section belongs. Note that the Content Engine's import subsystem does not tolerate unusually deep section hierarchies. Section trees that are more than 40 sections deep cannot be guaranteed to import successfully.

Syntax

```
<section
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    name="text"?
    unique-name="text"?
    mirror-source="(true|false)"?
  >
    <delete/>?
    <parent/>?
    <mirror-source/>?
    <unique-name>...</unique-name>?
    <directory>...</directory>?
    <section-layout>...</section-layout>?
    <article-layout>...</article-layout>?
    <priority/>?
  </section>
```

Examples

- This example imports a section. The child **parent** element specifies where the section is to be inserted into the section hierarchy. The referenced section must already exist in the database or appear before this element in the syndication file.

```
<section source="ex" sourceid="s2" name="example-section">
  <parent unique-name="ece_incoming"/>
</section>
```

- This example imports a section that may be mirrored by other sections.

```
<section source="ex" sourceid="s4" name="example-mirror-source" mirror-source="true">
  <parent unique-name="ece_incoming"/>
</section>
```

- This example imports a mirror section and illustrates the use of the **mirror-source** element to reference the section that is to be mirrored. The referenced section must already exist in the database or appear before this element in the syndication file.

```
<section source="ex" sourceid="s5" name="example-mirror-target" mirror-source="true">
  <parent unique-name="ece_incoming"/>
  <mirror-source unique-name="example-mirror-source"/>
</section>
```

Attributes

`id="`*`text`*`" (optional)`

A unique identifier for this **`section`** element. It is only valid and unique within the current syndication file and can be used to enable the establishment of relationships between elements in the file. Other elements in the file have **`id-ref`** attributes that can be used to reference **`section`** elements. If a **`section`** element does not have an **`id`** attribute then it must have either a **`dbid`** attribute or both a **`source`** and a **`sourceid`** attribute. A **`section`** element **may** have several or all of these attributes, in which case any of them can be used for establishing relationships.

The **`id`** attribute is not imported along with sections. Unless a **`dbid`** attribute has been specified, all imported sections are assigned new internal IDs during import.

`source="`*`text`*`" (optional)`

The name of the system from which this section originates. Together with the **`sourceid`** attribute it forms a globally unique external identifier for the section that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **`source`** and **`sourceid`** attributes that can be used for this purpose. If this attribute is specified then a **`sourceid`** attribute must also be specified. If a **`section`** element does not have a **`source`** and **`sourceid`** attribute then it must have either a **`dbid`** attribute or an **`id`** attribute. A **`section`** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If supplied, **`source`** and **`sourceid`** are imported and stored with sections. If **`source`** and **`sourceid`** are supplied and **`dbid`** is not supplied, then they are used to lookup an existing section. If a section with matching **`source`** and **`sourceid`** is found, then this section is updated; otherwise a new section is created.

If supplied, **`source`** and **`sourceid`** are imported and stored when creating new sections, but not when updating existing sections.

`sourceid="`*`text`*`" (optional)`

The id of this section in the system from which it originates. Together with the **`source`** attribute it forms a globally unique external identifier for the **`section`** that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **`source`** and **`sourceid`** attributes that can be used for this purpose. If this attribute is specified then a **`source`** attribute must also be specified. If a **`section`** element does not have a **`source`** and **`sourceid`** attribute then it must have either a **`dbid`** attribute or an **`id`** attribute. A **`section`** element

may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If **source** and **sourceid** are supplied and **dbid** is not supplied, then they are used to lookup an existing section. If a section with matching **source** and **sourceid** is found, then this section is updated; otherwise a new section is created.

If supplied, **source** and **sourceid** are imported and stored when creating new sections, but not when updating existing sections.

dbid="*text*" (optional)
The internal Content Engine ID of this section, which can be used when importing updated versions of existing content items. It can also be used for establishing relationships between elements in the syndication file. Other elements in the file have **dbid** attributes that can be used for this purpose. If a **section** element does not have a **dbid** attribute then it must have either a **source** and **sourceid** attribute or an **id** attribute. A **section** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

You should only use the **dbid** attribute when importing updated versions of **existing** sections.

This attribute is never present in syndication files that have been exported from a database. The ID is always written to the **exported-dbid** attribute in exported syndication files.

---

exported-dbid="*text*" (optional)
The internal Content Engine ID of this section, which can be used to identify the section in the database from which it was exported.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

---

name="*text*" (optional)
The name of this section.

unique-name="*text*" (optional)
The **unique-name** or **name** of an existing section to be updated. You can only use this attribute for look-up purposes, not for setting a section's unique name. To set the unique name of a section you are creating or updating, use the child **unique-name** element.

If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a **uniquename** or **name** attribute that matches this attribute, **or**

- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **section** element in the syndication file.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

mirror-source="(true|false)" (optional)
If **true**, then this section may be mirrored. This attribute may not be set to true if the section has a child **mirror-source** element.

5.33 section-acl

Represents an Escenic **access control list (ACL)**, which assigns a specified **role** to one or more users or user groups. A role implies a defined set of access rights. This element represents a **section** ACL, so the access rights only applies to a specified section of the publication.

Syntax

```
<section-acl
    name="(reader|administrator|useradmin|editor|journalist)"
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
    unique-name="text"?
  >
    <user-ref/>*
    <user-group-ref/>*
  </section-acl>
```

Attributes

name="(reader|administrator|useradmin|editor|journalist)"
The name of the **role** represented by this ACL.

Allowed values are:

reader

administrator

useradmin

editor

journalist

id-ref="*text*" (optional)
The **id** of the section to which this ACL is to apply. If this attribute is specified, a **section** element with an **id** attribute that matches this attribute must appear somewhere **before** this **section-acl** element in the syndication file.

If the section can not be found using **dbid** or **unique-name** or **source** and **sourceid** or any of those attributes is not specified, then this attribute will be used to find the section.

source="*text*" (optional)
The **source** of the section to which this ACL is to apply. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **section-acl** element in the syndication file.

If the section can not be found using **dbid** or the **dbid** attribute is not specified, then **source** and **sourceid** attributes are used to find the section.

sourceid="*text*" (optional)
The **sourceid** of the section to which this ACL is to apply. If this attribute is specified, then **source** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **section-acl** element in the syndication file.

If the section can not be found using **dbid** or the **dbid** attribute is not specified , then **source** and **sourceid** attributes are used to find the section.

dbid="*text*" (optional)
The **dbid** of the section to which this ACL is to apply. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a **dbid** attribute that matches this attribute, **or**
- A **section** element with a **dbid** attribute that matches this attribute must appear somewhere **before** this **section-acl** element in the syndication file.

exported-dbid="*text*" (optional)
The **dbid** of the section to which this ACL is to apply.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

---

unique-name="*text*" (optional)
The **unique-name** or **name** of the section to which this ACL is to apply. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a **uniquename** or **name** attribute that matches this attribute, **or**
- A **section** element with a **unique-name** or **name** attribute that matches this attribute must appear somewhere **before** this **section-acl** element in the syndication file.

If this is not the case, or if there is a matching **name** attribute but it is not unique, then import will fail.

If the section can not be found using **dbid** or **source** and **sourceid** or those attributes are not specified, then this attribute will be used to find the section.

5.34 section-layout

The name of the section layout to use for the section.

Syntax

```
<section-layout>
    text
  </section-layout>
```

5.35 section-page

Represents a section page in an Escenic publication. A section page contains links (called **content-refs**) to a selection of the content items in its owning section. The layout of the links on a section page is determined by layout objects defined in the **layout-group** publication resource. These objects are called **groups** and **areas**.

A section may have more than one section page, but only one of them is active at any given time.

A syndication file **section-page** element has a **layout-name** attribute that references a **group** element in the **layout-group** resource. This **group** defines the page's root group, and thus determine its layout. It also has child **area** elements that are used to hold the **content-ref** elements representing the links that are to appear on the page. The value assigned to the **layout-name** attribute determines what child **area** elements the **section-page** element may contain.

Syntax

```
<section-page
    name="text"?
    id-ref="text"?
    (source="text" sourceid="text")?
    dbid="text"?
```

```
  exported-dbid="text"?
  unique-name="text"?
  action="(remove|replace)"?
  layout-name="text"?
>
  <options>...</options>?<area>...</area>+
</section-page>
```

Examples

- This example imports a section page. The **source** and **source-id** attributes identify the section to which the section page is to be added. This section must already exist in the database or appear before this element in the syndication file. The **layout-name** attribute specifies the name of a root group defined in the publication's **layout-group** resource. The names of the descendant **area** and **group** elements reference members of this group as defined in the **layout-group** resource.

```
<section-page source="ex" sourceid="s12" name="example-section-8" layout-name="news">
  <area name="center">
    <group name="twoCol">
      <area name="left">
        <list-ref source="ex" sourceid="s12" name="important" number-of-items="2"/>
      </area>
      <area name="right">
        <content-ref source="ex" sourceid="19">
          <field name="leadtext">New lead text</field>
        </content-ref>
        <content-ref source="ex" sourceid="20"/>
      </area>
    </group>
  </area>
</section-page>
```

Attributes

name="*text*" (optional)
The name of this section page.

id-ref="*text*" (optional)
The **id** of the section to which this section page is to be added. If this attribute is specified, a **section** element with an **id** attribute that matches this attribute must appear somewhere **before** this **section-page** element in the syndication file.

If **dbid** or **source** and **sourceid** are specified, then this attribute is ignored.

source="*text*" (optional)
The **source** of the section to which this section page is to be added. If this attribute is specified, then **sourceid** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **source** and **sourceid** attributes that match **source** and **sourceid**, **or**
- A **section** element with **source** and **sourceid** attributes that match **source** and **sourceid** must appear somewhere **before** this **section-page** element in the syndication file.

If **dbid** is specified, then **source** and **sourceid** are ignored.

`sourceid="`*`text`*`" (optional)`
The **`sourceid`** of the section to which this section page is to be added. If this attribute is specified, then **`source`** must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with **`source`** and **`sourceid`** attributes that match **`source`** and **`sourceid`**, **or**
- A **`section`** element with **`source`** and **`sourceid`** attributes that match **`source`** and **`sourceid`** must appear somewhere **before** this **`section-page`** element in the syndication file.

If **`dbid`** is specified, then **`source`** and **`sourceid`** are ignored.

`dbid="`*`text`*`" (optional)`
The **`dbid`** of the section to which this section page is to be added. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a **`dbid`** attribute that matches this attribute, **or**
- A **`section`** element with a **`dbid`** attribute that matches this attribute must appear somewhere **before** this **`section-page`** element in the syndication file.

`exported-dbid="`*`text`*`" (optional)`

`unique-name="`*`text`*`" (optional)`
The **`unique-name`** or **`name`** of the section to which this section page is to be added. If this attribute is specified, then one of the following conditions must be satisified:

- The target publication must already contain a section with a **`uniquename`** or **`name`** attribute that matches this attribute, **or**
- A **`section`** element with a **`unique-name`** or **`name`** attribute that matches this attribute must appear somewhere **before** this **`section-page`** element in the syndication file.

If this is not the case, or if there is a matching **`name`** attribute but it is not unique, then import will fail.

If **`dbid`** or **`source`** and **`sourceid`** or **`id`** are specified, then this attribute is ignored.

`action="(remove|replace)" (optional)`
Determines what action is taken during import if the section page already exists.

Allowed values are:

`remove`
All the section page's existing areas are removed and the new areas imported.

`replace` (default)
Imported areas replace any existing areas with the same name; other existing areas are not removed.

`layout-name="`*`text`*`"` (optional)
The name of a group defined in the **`layout-groups`** resource. The selected group will be the section page's root group and thus determine the layout of the section page. The group you specify here determines what **`area`** elements this **`section-page`** element may contain.

5.36 section-ref

A reference to a section to which this element's owning content item (**`content`** element) belongs. Since a content item can belong to many sections, a **`content`** element may contain many **`section-ref`** elements. A section can be referenced using any of the following attributes:

- **`id-ref`**
- **`source`** and **`source-id`**
- **`dbid`**
- **`unique-name`**

Import considerations

The referenced section must either be defined prior to this element in the syndication file or else already exist in the target publication.

Syntax

```
<section-ref
    id-ref="text"?
    publication-name="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    unique-name="text"?
    todesk="(true|false)"?
    home-section="(true|false)"?
  />
```

Examples

- This example imports an "image" content item. It includes a reference to a section to which the content item is to belong. This section is set to be the content item's home section. The referenced section must already exist in the database or appear before this **`content`** element in the syndication file.

  Note the use of JSON syntax to include soft crop information in the **`representations`** field. For more information about this, see **section 4.1**.

```
<content source="ex" sourceid="20" type="picture" state="published">
  <section-ref source="ex" sourceid="s2" home-section="true"/>
  <field name="title">Croc</field>
  <field name="caption">A Croc on the beach</field>
  <field name="binary" title="crocodile">/tmp/escenic/import/croc.jpg</field>
  <field name="representations">
      {
        "thumbnail":
          {
```

```
            "crop":
              {
                "width":400,
                "height":400,
                "x":0,
                "y":54
              }
          },
        "narrow":
          {
            "crop":
              {
                "width":250,
                "height":200,
                "x":0,
                "y":54
              }
          },
        "wide":
          {
            "crop":
              {
                "width":400,
                "height":400,
                "x":102,
                "y":100
              }
          }
      }
    </field>
</content>
```

- This example imports a content item and inserts it into four different sections, two of which belong to a different publication. One of the local sections is set to be the content item's home section. In addition, one of the remote sections is set to be the content item's home section in that publication.

```
<content source="ex" sourceid="99" type="news" state="published">
  <section-ref unique-name="sports" home-section="true"/>
  <section-ref unique-name="ece_frontpage"/>
  <section-ref unique-name="sports" home-section="true" publication-name="other"/>
  <section-ref unique-name="ece_frontpage" publication-name="other"/>
  <field name="title">Local Cup Fiasco</field>
  <field name="leadtext">...</field>
  <field name="body">...</field>
</content>
```

Attributes

`id-ref="`*`text`*`"` (optional)
The **`id`** of the section to which the content item represented by the owning **`content`** element belongs/is to be added. If this attribute is specified, a **`section`** element with an **`id`** attribute that matches this attribute must appear somewhere **before** this **`section-ref`** element in the syndication file.

If **`dbid`** or **`source`** and **`sourceid`** are specified, then this attribute is ignored.

`publication-name="`*`text`*`"` (optional)
The name of the publication to which the referenced content item or section belongs. This attribute may only be used in combination with the **`source`** and **`source-id`** attributes. It is needed to ensure unique identification in situations where cross-publishing is in use and the referenced content item or section does not belong to the current publication.

`source="`*text*`"` (optional)
The `source` of the section to which the content item represented by the owning `content` element belongs/is to be added. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with `source` and `sourceid` attributes that match `source` and `sourceid`, **or**
- A `section` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `section-ref` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`sourceid="`*text*`"` (optional)
The `sourceid` of the section to which the content item represented by the owning `content` element belongs/is to be added. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The target publication must already contain a section with `source` and `sourceid` attributes that match `source` and `sourceid`, **or**
- A `section` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `section-ref` element in the syndication file.

If `dbid` is specified, then `source` and `sourceid` are ignored.

`dbid="`*text*`"` (optional)
The `dbid` of the section to which the content item represented by the owning `content` element belongs/is to be added. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a section with a `dbid` attribute that matches this attribute, **or**
- A `section` element with a `dbid` attribute that matches this attribute must appear somewhere **before** this `section-ref` element in the syndication file.

`unique-name="`*text*`"` (optional)
The `unique-name` or `name` of the section to which the content item represented by the owning `content` element belongs/is to be added. If this attribute is specified, then one of the following conditions must be satisfied:

- The target publication must already contain a section with a `uniquename` or `name` attribute that matches this attribute, **or**
- A `section` element with a `unique-name` or `name` attribute that matches this attribute must appear somewhere **before** this `section-ref` element in the syndication file.

If this is not the case, or if there is a matching `name` attribute but it is not unique, then import will fail.

If **dbid** or **source** and **sourceid** or **id** are specified, then this attribute is ignored.

todesk="(true|false)" (optional)
If set to **true**, then content item represented by the owning **content** element is added to this section's default inbox (**INBOX**).

home-section="(true|false)" (optional)
If set to **true** then this section is the home section of the content item represented by the owning **content** element. If the **publication-name** attribute is also specified, then this section is the content item's local home section in the named publication.

5.37 tag

Either:

- a tag to be attached to the content item represented by this element's parent **content** element.
- an instruction to remove all tags currently attached to the content item represented by this element's parent **content** element.

Syntax

```
<tag
    ( action="(remove)" | identifier="text" relevance="text" )
  />
```

Attributes

action="(remove)"
If this attribute is specified, all tags currently attached to the content item represented by this element's parent **content** element will be removed.

Allowed values are:

remove

identifier="*text*"
A tag identifier of the form:

```
scheme-uri:term
```

where:

scheme-uri
is the scheme URI of one of the tag structures defined at your installation. The **escenic-admin** web application's **Manage Tag Structures** option displays a list of all available tag structures and their scheme URIs.

term
is the local identifier of one of the tags in the tag structure identified by *scheme-uri*. If you don't know the terms of the tags in a tag structure, you can access them using the web service. See the **Integration Guide** for details.

`relevance="`*text*`"`
Defines the relevance between this tag and this element's parent **`content`** element.

This attribute is optional

5.38 unique-name

A unique name to be assigned to the section created or updated by this element's parent **`section`** element.

Syntax

```
<unique-name>
    text
  </unique-name>
```

5.39 update

When importing a content item that already exists in the target publication, you can use this element to update the content item's source and source ID references.

Syntax

```
<update
    newsource="text"?
    newsourceid="text"?
  />
```

Attributes

`newsource="`*text*`"` (optional)
The new source name to be assigned to the content item in the target publication.

`newsourceid="`*text*`"` (optional)
The new source ID to be assigned to the content item in the target publication.

5.40 uri

The URI you want to be assigned to a content item. The content item's URI will be formed by appending the value you specify here to the URI of the content item's home section, so you must specify a relative URI. If you do not specify this element, then the content item will be assigned an automatically generated URI.

Syntax

```
<uri
    use-as-default="text"?
  >
    text
  </uri>
```

Attributes

`use-as-default="`*text*`"` (optional)
If use-as-default is set to false, then the uri is added as an alias for the content item. The uri can then be used to look up the content item but asking a content item for its url will not return this value. There can only be one default uri for a content item. Specifying a new default will replace the old default uri. The old default uri value will be kept as an alias.

5.41 user

Represents a special kind of content item used to store user information about users of a publication.

Syntax

```
<user
    id="text"?
    (source="text" sourceid="text")?
    dbid="text"?
    exported-dbid="text"?
  >
    <field>...</field>*
  </user>
```

Child Elements

`field`: **section 5.10**.

Only one form of the **`field`** element may be used: **User field** (**section 5.10.3**).

Attributes

`id="`*text*`"` (optional)
A unique identifier for this **`user`** element. It is only valid and unique within the current syndication file and can be used to enable the establishment of relationships between elements in the file. Other elements in the file have **`id-ref`** attributes that can be used to reference **`user`** elements. If a **`user`** element does not have an **`id`** attribute then it must have either a **`dbid`** attribute or both a **`source`** and a **`sourceid`** attribute. A **`user`** element **may** have several or all of these attributes, in which case any of them can be used for establishing relationships.

The **`id`** attribute is not imported along with users. Unless a **`dbid`** attribute has been specified, all imported users are assigned new internal IDs during import.

`source="`*`text`*`"` (optional)
The name of the system from which this user originates. Together with the **`sourceid`** attribute it forms a globally unique external identifier for the user that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **`source`** and **`sourceid`** attributes that can be used for this purpose. If this attribute is specified then a **`sourceid`** attribute must also be specified. If a **`user`** element does not have a **`source`** and **`sourceid`** attribute then it must have either a **`dbid`** attribute or an **`id`** attribute. A **`user`** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If supplied, **`source`** and **`sourceid`** are imported and stored with users. If **`source`** and **`sourceid`** are supplied and **`dbid`** is not supplied, then they are used to lookup an existing user. If a user with matching **`source`** and **`sourceid`** is found, then this user is updated; otherwise a new user is created.

If supplied, **`source`** and **`sourceid`** are imported and stored when creating new users, but not when updating existing users.

`sourceid="`*`text`*`"` (optional)
The id of this user in the system from which it originates. Together with the **`source`** attribute it forms a globally unique external identifier for the **`user`** that can be used for establishing relationships between elements in the syndication file. Other elements in the file have **`source`** and **`sourceid`** attributes that can be used for this purpose. If this attribute is specified then a **`source`** attribute must also be specified. If a **`user`** element does not have a **`source`** and **`sourceid`** attribute then it must have either a **`dbid`** attribute or an **`id`** attribute. A **`user`** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

If **`source`** and **`sourceid`** are supplied and **`dbid`** is not supplied, then they are used to lookup an existing user. If a user with matching **`source`** and **`sourceid`** is found, then this user is updated; otherwise a new user is created.

If supplied, **`source`** and **`sourceid`** are imported and stored when creating new users, but not when updating existing users.

`dbid="`*`text`*`"` (optional)
The internal Content Engine ID of this user, which can be used when importing updated versions of existing content items. It can also be used for establishing relationships between elements in the syndication file. Other elements in the file have **`dbid`** attributes that can be used for this purpose. If a **`user`** element does not have a **`dbid`** attribute then it must have either a **`source`** and **`sourceid`** attribute or an **`id`** attribute. A **`user`** element may have several or all of these attributes, in which case any of them can be used for establishing relationships.

You should only use the **dbid** attribute when importing updated versions of **existing** users.

This attribute is never present in syndication files that have been exported from a database. The ID is always written to the **exported-dbid** attribute in exported syndication files.

---

exported-dbid="*text*" (optional)

The internal Content Engine ID of this user, which can be used to identify the user in the database from which it was exported.

This attribute is generated during export from the , but ignored during import. It is provided mainly for information and debugging purposes.

5.42 user-group

Represents an Escenic user group. User groups are a convenient means of managing access rights: roles can be assigned to whole groups of users rather than to individuals.

Syntax

```
<user-group
    id="text"?
    name="text"
    publication-id="integer"?
    publication-name="..."?
  >
    <user-ref/>*
    <user-group-ref/>*
  </user-group>
```

Attributes

id="*text*" (optional)

A unique identifier for this **user-group** element. It is only valid and unique within the current syndication file and can be used to enable the establishment of relationships between elements in the file. Other elements in the file have **id-ref** attributes that can be used to reference **user-group** elements. If a **user-group** element does not have an **id** attribute then it must have either a **dbid** attribute or both a **source** and a **sourceid** attribute. A **user-group** element **may** have several or all of these attributes, in which case any of them can be used for establishing relationships.

The **id** attribute is not imported along with user-groups. Unless a **dbid** attribute has been specified, all imported user-groups are assigned new internal IDs during import.

name="*text*"

The name of this user-group.

publication-id="*integer*" (optional)
The ID of the publication to which this **user-group** belongs.

publication-name="..." (optional)
The name of the publication to which this **user-group** belongs.

5.43 user-group-ref

References a user group to which the role represented by this element's owning ACL is to be assigned.

Syntax

```
<user-group-ref
    id-ref="text"?
    name="text" publication-id="integer"?
  />
```

Attributes

id-ref="*text*" (optional)
The **id** of the referenced user group. If this attribute is specified, a **user-group** element with an **id** attribute that matches this attribute must appear somewhere **before** this **user-group-ref** element in the syndication file.

If **name** and **publication-id** are specified, then this attribute is ignored.

name="*text*" (optional)
The name of the referenced user group. If this attribute is specified, then **publication-id** must also be specified. One of the following two conditions must be satisfied:

- The publication specified with **publication-id** must contain a user group with this name.
- A **user-group** element with matching **publication-id** and **name** attributes must appear somewhere **before** this **user-group-ref** element in the syndication file.

publication-id="*integer*" (optional)
The ID of the publication containing the referenced user group. If this attribute is specified, then **name** must also be specified. One of the following two conditions must be satisfied:

- The specified publication must contain a user group with the name specified with the **name** attribute.
- A **user-group** element with matching **publication-id** and **name** attributes must appear somewhere **before** this **user-group-ref** element in the syndication file.

5.44 user-ref

References a user to which the role represented by this element's owning ACL is to be assigned.

Syntax

```
<user-ref
    id-ref="text"?
    (source="text" sourceid="text")?
    username="text"?
    dbid="text"?
    exported-dbid="text"?
  />
```

Attributes

`id-ref="`*`text`*`"` (optional)
The `id` of the referenced user. If this attribute is specified, a `user` element with an `id` attribute that matches this attribute must appear somewhere **before** this `user-ref` element in the syndication file.

If the user can not be found using `dbid` or `username` or `source` and `source-id` or those attributes are not specified, then this attribute is used to find the user.

`source="`*`text`*`"` (optional)
The `source` of the referenced user. If this attribute is specified, then `sourceid` must also be specified. One of the following two conditions must be satisfied:

- The database must already contain a user with `source` and `sourceid` attributes that match `source` and `sourceid`, **or**
- A `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `user-ref` element in the syndication file.

If the user can not be found using the `dbid` attribute or the `dbid` attribute is not specified, then `source` and `sourceid` attributes will be used.

`sourceid="`*`text`*`"` (optional)
The `sourceid` of the referenced user. If this attribute is specified, then `source` must also be specified. One of the following two conditions must be satisfied:

- The database must already contain a user with `source` and `sourceid` attributes that match `source` and `sourceid`, **or**
- A `user` element with `source` and `sourceid` attributes that match `source` and `sourceid` must appear somewhere **before** this `user-ref` element in the syndication file.

If the user can not be found using the `dbid` attribute or the `dbid` attribute is not specified, then `source` and `sourceid` attributes will be used.

`username="`*`text`*`"` (optional)
The username of the referenced user. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a "person" or "user" content item with a **`username`** field that matches this attribute, **or**
- A **`user`** element with a field called **`username`** that matches this attribute must appear somewhere **before** this **`user-ref`** element in the syndication file.

If the user can not be found using the **`dbid`** or **`source`** and **`sourceid`** or those attributes are not specified, then **`username`** attribute will be used to find the user.

`dbid="`*`text`*`"` (optional)
The **`dbid`** of the referenced user. If this attribute is specified then one of the following two conditions must be satisfied:

- The target publication must already contain a user with a **`dbid`** attribute that matches this attribute, **or**
- A **`user`** element with a **`dbid`** attribute that matches this attribute must appear somewhere **before** this **`user-ref`** element in the syndication file.

`exported-dbid="`*`text`*`"` (optional)
The **`dbid`** of the referenced user.

This attribute is generated during export from the Content Engine, but ignored during import. It is provided mainly for information and debugging purposes.

5.45 value

A single value within a **`field`**. A **`field`** element may contain a series of **`value`** elements if it is defined in the **`content-type`** resource as having the type **`array`** or **`enumeration`**.

If the field is an array, then each **`value`** element represents an element of the array and can either contain a simple text value or a **`field`** element if it is a complex array.

If the field is an enumeration, then each **`value`** element represents one of the possible values to which the field can be set and must contain a simple text value.

Syntax

```
<value>
    (text|<field>...</field>*|<schedule:schedule>...</schedule:schedule>)
  </value>
```

Child Elements

text, `field`: **section 5.10**, `schedule:schedule`: **section 5.30**.

Only one form of the `field` element may be used: **Standard field** (**section 5.10.2**).