Media Field Type

This Field Type represents and handles media (audio/video) binary file.

Name	Internal name	Expected input
Media	ezmedia	mixed

Description

This Field Type represents and handles a media (audio/video) binary file.

It is capable of handling following types of files:

- Apple QuickTime
- Adobe Flash
- Microsoft Windows Media
- Real Media
- Silverlight
- HTML5 Video
- HTML5 Audio

PHP API Field Type

Input expectations

Type	Description	Example
string	Path to the media file.	/Users/jane/butterflies.mp4
eZ\Publish\Core\FieldType\Media\Value	Media FieldType Value Object with path to the media file as the value of id property.	See **Value** object section below.

Value object

Properties

eZ\Publish\Core\FieldType\Media\Value offers the following properties.

> Note that both **Media** and **BinaryFile** Value and Type inherit from the **BinaryBase** abstract Field Type and share common properties.

Property	Type	Description	Example
id	string	Media file identifier. This ID depends on the IO Handler that is being used. With the native, default handlers (FileSystem and Legacy), the ID is the file path, relative to the binary file storage root dir (var/<vardir>/storage/original by default).	application/63cd472dd7819da7b75e8e2fee507c68.mp4

fileName	string	The human-readable file name, as exposed to the outside. Used when sending the file for download in order to name the file.	butterflies.mp4
fileSize	int	File size, in bytes.	1077923
mimeType	string	The file's mime type.	video/mp4
uri	string	The binary file's HTTP uri. If the URI doesn't include a host or protocol, it applies to the request domain. **The URI is not publicly readable, and must NOT be used to link to the file for download.** Use `ez_render_field` to generate a valid link to the download controller.	/var/ezdemo_site/storage/original/application/63cd472dd7819da7b75e8e2fee507c68.mp4
hasController	boolean	Whether the media has a controller when being displayed.	true
autoplay	boolean	Whether the media should be automatically played.	true
loop	boolean	Whether the media should be played in a loop.	false
height	int	Height of the media.	300
width	int	Width of the media.	400
path	string	**deprecated** Renamed to `id` starting from eZ Publish 5.2. Can still be used, but it is recommended not to use it anymore as it will be removed.	

Hash format

The hash format mostly matches the value object. It has the following keys:

- `id`
- `path` (for backwards compatibility)
- `fileName`
- `fileSize`
- `mimeType`
- `uri`
- `hasController`
- `autoplay`
- `loop`
- `height`
- `width`

Validation

The Field Type supports **FileSizeValidator**, defining maximum size of media file in bytes:

Name	Type	Default value	Description
maxFileSize	int	false	Maximum size of the file in bytes.

Example of using Media Field Type validator in PHP

```php
use eZ\Publish\Core\FieldType\Media\Type;

$contentTypeService =
$repository->getContentTypeService();
$mediaFieldCreateStruct =
$contentTypeService->newFieldDefinitionCreateStruct(
"media", "ezmedia" );

// Setting maximum file size to 5 megabytes
$mediaFieldCreateStruct->validatorConfiguration = array(
    "FileSizeValidator" => array(
        "maxFileSize" => 5 * 1024 * 1024
    )
);
```

Settings

The Field Type supports `mediaType` setting, defining how the media file should be handled in output.

Name	Type	Default value	Description
mediaType	mixed	Type::TYPE_HTML5_V IDEO	Type of the media, accepts one of the predefined constants.

List of all available `mediaType` constants defined in `eZ\Publish\Core\FieldType\Media\Type` class:

Name	Description
TYPE_FLASH	Adobe Flash
TYPE_QUICKTIME	Apple QuickTime
TYPE_REALPLAYER	Real Media
TYPE_SILVERLIGHT	Silverlight
TYPE_WINDOWSMEDIA	Microsoft Windows Media
TYPE_HTML5_VIDEO	HTML5 Video
TYPE_HTML5_AUDIO	HTML5 Audio

Example of using Media Field Type settings in PHP

```php
use eZ\Publish\Core\FieldType\Media\Type;

$contentTypeService =
$repository->getContentTypeService();
$mediaFieldCreateStruct =
$contentTypeService->newFieldDefinitionCreateStruct(
"media", "ezmedia" );

// Setting Adobe Flash as the media type
$mediaFieldCreateStruct->fieldSettings = array(
    "mediaType" => Type::TYPE_FLASH,
);
```