Asset Management

eZ Platform supports multiple binary file handling mechanisms by means of an `IOHandler` interface. This feature is used by the BinaryFile , Media and Image Field Types.

In this topic:

Native IO handler

The IO API is organized around two types of handlers:

- `eZ\Publish\IO\IOMetadataHandler`: Stores & reads metadata (validity, size, etc.)
- `eZ\Publish\IO\IOBinarydataHandler`: Stores & reads binarydata (actual contents)

The IOService uses both.

Configuration

IO handling can now be configured using semantic configuration. Assigning the IO handlers to ezplatform itself is configurable per siteaccess. This is the default configuration:

```
ezpublish:
    system:
        default:
            io:
                metadata_handler: default
                binarydata_handler: default
```

Metadata and binarydata handlers are configured in the `ez_io` extension. This is what the configuration looks like for the default handlers. It declares a metadata handler and a binarydata handler, both labelled 'default'. Both handlers are of type 'flysystem', and use the same flysystem adapter, labelled 'default' as well.

```
ez_io:
    metadata_handlers:
        default:
            flysystem:
                adapter: default
    binarydata_handlers:
        default:
            flysystem:
                adapter: default
```

The 'default' flysystem adapter's directory is based on your site settings, and will automatically be set to `%ezpublish_legacy.root_dir%/$var_dir$/$storage_dir$` (example: `/path/to/ezpublish_legacy/var/ezdemo_site/storage`).

Configure the permissions of generated files

V1.5

```
ezpublish:
    system:
        default:
            io:
                permissions:
                    files: 0750 #default is 0644
                    directories: 0640 #default is 0755
```

Both `files` and `directories` rules are optional.

Default values are 0644 for files and 0755 for directories.

The native Flysystem handler

league/flysystem (along with FlysystemBundle) is an abstract file handling library.

It is used as the default way to read & write content binary files in eZ Platform. It can use the `local` filesystem *(our default configuration and the one we officially support)*, but is also able to read/write to `sftp`, `zip` or cloud filesystems *(`azure`, `rackspace`, S3)*.

Handler options

Adapter

The adapter is the 'driver' used by flysystem to read/write files. Adapters can be declared using `oneup_flysystem` as follows:

```
oneup_flysystem:
    adapters:
        default:
            local:
                directory: "/path/to/directory"
```

The way to configure other adapters can be found on the bundle's online documentation. Note that we do not use the Filesystem configuration described in this documentation, only the adapters.

The DFS Cluster handler

For clustering use we provide a custom metadata handler that stores metadata about your assets in the database. This is done as it is faster then accessing the remote NFS or S3 instance in order to read metadata. For further reading on setting this up, see Clustering.