Binary files handling

The eZ Publish 5 API supports multiple binary file handling mechanisms by means of an `IOHandler` interface. This feature is used by the Binary File, Media and Image FieldTypes.

Starting from versions 5.4 / 2014.09, a native IO handler is available.

> The LegacyKernel IO handler, that runs legacy kernel callbacks, is removed.

Native IO handler 5.4 / 2014.09

The unique `IOHandler` interface is removed, replace by two distinct interfaces:

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

The IOService uses both.

Configuration

IO handling can now be configured using semantic configuration.

Which IO handlers (metadata & binarydata) is configurable by 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`).

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 Publish. It can use the local filesystem (our default configuration), but is also able to read/write to sftp, zip or cloud filesystems (dropbox, rackspace, aws-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"
```

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

Upgrading

For those using the default `eZFSFileHandler`, no configuration should be required, and things should just work like before, but without legacy kernel callbacks for manipulating images & binary files.

For those using the DFS cluster file handler, a new native handler is available. See the DFS specification.