aaf2.file module

class **aaf2.file.AAFFactory**(*root*)

Bases: `object`

from_name(*name*, **args*, ***kwargs*)

create_instance(**args*, ***kwargs*)

class **aaf2.file.AAFObjectManager**(*root*)

Bases: `object`

create_temp_dir()

remove_temp()

add_modified(*obj*)

pop(*path*, *default=None*)

read_object(*path*)

write_objects()

class **aaf2.file.AAFFile**(*path=None*, *mode='r'*, *sector_size=4096*, *extensions=True*, *buffering=8192*)

Bases: `object`

AAF File Object. This is the entry point object for most of the API. This object is designed to be like python's native open function. It is recommended to create this object with the *aaf.open* alias. It is also highly recommended to use the with statement.

Warning

If an exception is raised inside the with block and the file was opened as writable, the final file should be considered bad or corrupted.

Take this snippet as an example:

```
try:
    with aaf.open('/path/to/aaf_file.aaf', 'r+') as f:
        raise ValueError('asd')
except:
    pass
```

in this case, even if the exception is properly handled, the content of `/path/to/aaf_file.aaf` shouldn't be trusted anymore.

For example. Opening existing AAF file readonly:

```
with aaf.open('/path/to/aaf_file.aaf', 'r') as f:
```

Opening new AAF file overwriting existing one:

```
with aaf.open('/path/to/aaf_file.aaf', 'w') as f:
```

Opening existing AAF in read and write:

```
with aaf.open('/path/to/aaf_file.aaf', 'rw') as f:
```

Opening in memory BytesIO file:

```
with aaf.open() as f:
```

header

`aaf2.content.Header` object for AAF file.

content

`aaf2.content.ContentStorage` object for AAF File. This has the Mob and EssenceData objects.

dictionary

`aaf2.dictionary.Dictionary` for AAF file. The dictionary property has DefinitionObject objects.

setup_empty()

writeable

resovle_weakref(*index*, *ref_pid*, *ref*)

weakref_prop(*index*)

weakref_index(*pid_path*)

read_reference_properties()

write_reference_properties()

dump()

save()

Writes current changes to disk and flushes modified objects in the AAFObjectManager

close()

Close the file. A closed file cannot be read or written any more.