
mosromgr

Release 0.9.1

unknown

Sep 02, 2021

CONTENTS

1	Example Usage	3
2	Documentation	5
3	Issues and questions	67
4	Contributing	69
5	Contributors	71
6	Licence	73
7	Contact	75
	Python Module Index	77
	Index	79

Python library for managing **MOS** running orders. Pronounced *mos-ro-manager*.



The library provides functionality for classifying MOS file types, processing and inspecting MOS message files, as well as merging MOS files into a running order, and providing a “completed” programme including all additions and changes made between the first message (`roCreate`) and the last (`roDelete`).

This can be used as a library, using the utilities provided in the *mosromgr* module, and the command line command *CLI* can be used to process either a directory of MOS files, or a folder within an S3 bucket.

This library was developed by the [BBC News Labs](#) team.

Warning: Note that the library is currently in beta. The API and CLI are not yet stable and may change. Once the library reaches v1.0, it will be considered stable. Please consider giving [Feedback](#) to help stabilise the API.

EXAMPLE USAGE

1.1 Command line

List the stories within a running order:

```
$ mosromgr inspect -f roCreate.mos.xml --stories
0828 MIDLANDS TODAY Wed, 11.11.2020

INTRODUCTION-READ

TESTING-OOV

WEATHER-SHORT

END OF PROGRAMME
```

Merge all MOS files in directory *newsnight* and save in *FINAL.xml*:

```
$ mosromgr merge -f newsnight/* -o FINAL.xml
```

1.2 Library

Load a roCreate file and view its stories:

```
from mosromgr.mostypes import RunningOrder

ro = RunningOrder.from_file('roCreate.mos.xml')

for story in ro.stories:
    print(story.slug)
```

Merge a single roStorySend (*StorySend*) into a roCreate (*RunningOrder*) and output the file to a new file:

```
from mosromgr.mostypes import RunningOrder, StorySend

ro = RunningOrder.from_file('roCreate.mos.xml')
ss = StorySend.from_file('roStorySend.mos.xml')

ro += ss
```

(continues on next page)

(continued from previous page)

```
with open('final.mos.xml', 'w') as f:
    f.write(str(ro))
```

If you're automating this process you won't necessarily know which MOS Type to use, so you can construct an object from the base class *MosFile* which will automatically classify your file:

```
>>> from mosromgr.mostypes import MosFile
>>> mf1 = MosFile.from_file('roCreate.mos.xml')
>>> mf1
<RunningOrder 1000>
>>> mf2 = MosFile.from_file('roStorySend.mos.xml')
>>> mf2
<StorySend 1001>
```

Using *MosCollection* will sort and classify multiple MOS types of all given files, allowing you to process a collection of MOS files within a complete or partially complete programme:

```
from mosromgr.moscollection import MosCollection

mos_files = ['roCreate.mos.xml', 'roStorySend.mos.xml', 'roDelete.mos.xml']
mc = MosCollection.from_files(mos_files)

mc.merge()
with open('final.mos.xml', 'w') as f:
    f.write(str(mc))
```


DOCUMENTATION

This documentation follows the [Diátaxis](#) system, so is split between four modes of documentation: tutorials, how-to guides, technical reference and explanation.

2.1 Getting started

This section shows you how to get started with *mosromgr*.

2.1.1 Installing

Install with pip:

```
$ pip install mosromgr
```

2.1.2 Command line interface check

After installing the module, a simple way to verify it's working is by using the [CLI](#). First of all, open a terminal and run the command *mosromgr* to be sure it's installed. You should see output like so:

```
$ mosromgr
optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

commands:
  {help,detect,inspect,merge}
  help                 Displays help about the specified command
  detect               Detect the MOS type of one or more files
  inspect              Inspect the contents of a roCreate file
  merge                Merge the given MOS files
```

Now start by obtaining the MOS files for a single complete programme. In a terminal window, enter the directory containing the MOS files and run the command *mosromgr detect* on a single roCreate file, for example:

```
$ mosromgr detect 123456-roCreate.mos.xml
123456-roCreate.mos.xml: RunningOrder
```

The output shows that it's identified the roCreate file as a *RunningOrder*. Try it with some other files to check it can correctly identify a *MosFile* subclass to represent the file.

2.1.3 Using the module in Python code

Now you’ve tested the ready-made command line program is working with your MOS file, try using the module in some custom Python code.

Open a Python shell and try creating a MOS object from your roCreate file:

```
>>> from mosromgr.mostypes import RunningOrder
>>> ro = RunningOrder.from_file('123456-roCreate.mos.xml')
>>> ro
<RunningOrder 123456>
```

This shows you’ve successfully loaded a MOS file and created a *RunningOrder* from it. The output shows the object representation (`__repr__`) which includes the class name and message ID (this is from the XML contents, not the filename).

The next page will walk through the functionality provided by the module.

2.2 Introduction

This section is a walkthrough of the contents of the module, intended to explain how *mosromgr* works and introduce the concepts.

2.2.1 MOS Types

The *MOS Types* section of the module provides a collection of classes for dealing with individual MOS messages. The classes provide easy access to some of the elements within a MOS file, such as a list of stories within a running order, the transmission time of a programme, or its duration.

For example, you can load a running order from a roCreate file, print the RO Slug and access some details:

```
>>> from mosromgr.mostypes import RunningOrder
>>> ro = RunningOrder.from_file('123456-roCreate.mos.xml')
>>> ro.ro_slug
'22:45 NEWSNIGHT 54D CORE Thu, 08.04.2021'
>>> ro.message_id
123456
>>> ro.start_time
datetime.datetime(2021, 4, 8, 21, 46, 30)
>>> ro.duration
970.0
>>> len(ro.stories)
10
```

In the case of MOS messages which contain a *change* to a running order, the relevant details are exposed, for example a *StoryInsert* includes access to the *source_stories* and *target_story*.

When dealing with merging *MosFile* objects, this is done by “adding” each file to the *RunningOrder* object by using the + operator:

```
>>> from mosromgr.mostypes import RunningOrder, StoryInsert
>>> ro = RunningOrder.from_file('123456-roCreate.mos.xml')
>>> ss = StoryInsert.from_file('123457-roStoryInsert.mos.xml')
```

(continues on next page)

(continued from previous page)

```
>>> len(ro.stories)
10
>>> ro += ss
>>> len(ro.stories)
11
```

2.2.2 MOS Elements

The *MOS Elements* part of the module provides a collection of classes used to provide easy access to certain elements within a *MosFile* object, such as a list of stories within a running order, and the items within a story:

```
from mosromgr.mostypes import RunningOrder

ro = RunningOrder.from_file('123456-roCreate.mos.xml')

print(ro.ro_slug)
for story in ro.stories:
    print(story.slug)
```

Here, `ro.stories` is a list of *Story* objects. Each story has its own set of accessible properties, such as the story's *duration*, *start_time*, *end_time*, *offset* and *items*:

```
>>> story = ro.stories[0]
>>> story.duration
180.0
>>> story.start_time
datetime.datetime(2021, 4, 8, 21, 46, 30)
>>> len(story.items)
3
```

Here, the story contains 3 items, each of these is an *Item* object.

2.2.3 MOS Collection

The *MOS Collection* part of the module provides a wrapper class *MosCollection* which stores references to specified MOS files, strings or S3 object keys so the *MosFile* objects can be recreated when needed rather than kept in memory. Rather than using the `+` operator, a *merge()* method is provided:

```
from mosromgr.moscollection import MosCollection

mc = MosCollection.from_s3(bucket_name=bucket_name, prefix=prefix)

mc.merge()
```

The next page will cover some example problems and solutions to show you how you can use *mosromgr* in practice.

2.3 How-to guide

This section is a series of helpful recipes for how to do things and solve particular problems with *mosromgr*.

Note: For simplicity, these examples deal with MOS messages read from local files, but *MosFile* and *MosCollection* objects can also be constructed using *from_string* and *from_s3*. Refer to *MOS Types* and *MOS Collection* for more information.

2.3.1 Merging MOS files

When dealing with merging *MosFile* objects, this is done by “adding” each file to the *RunningOrder* object by using the + operator:

```
>>> from mosromgr.mostypes import RunningOrder, StoryInsert
>>> ro = RunningOrder.from_file('123456-roCreate.mos.xml')
>>> si = StoryInsert.from_file('123457-roStoryInsert.mos.xml')
>>> len(ro.stories)
10
>>> ro += si
>>> len(ro.stories)
11
```

To parse and merge a collection of MOS files, you could create a list of files (or use *glob()*), let *MosFile* classify each file, then merge each of them into the *RunningOrder*:

```
from mosromgr.mostypes import MosFile
from glob import glob

files = glob('*.mos.xml')

ro, *mosfiles = sorted(MosFile.from_file(f) for f in files)

for mf in mosfiles:
    ro += mf
```

To access the final XML, simply print the *RunningOrder* object or access the *__str__*:

```
>>> print(ro)
<mos>
  <mosID>MOS ID</mosID>
  <messageID>1234567</messageID>
  ...
>>> s = str(ro)
>>> s
<mos>
  <mosID>MOS ID</mosID>
  <messageID>1234567</messageID>
  ...
```

2.3.2 Merging MOS files using MOSCollection

The *MosCollection* class provides a wrapper for operations dealing with a collection of MOS files as part of one programme. So to merge files like in the previous example, you could do the following instead:

```
from mosromgr.moscollection import MosCollection
from glob import glob

files = glob('*.mos.xml')
mc = MosCollection.from_files(files)

mc.merge()
```

To access the final XML, simply print the *MosCollection* object or access the `__str__`:

```
>>> print(mc)
<mos>
  <mosID>MOS ID</mosID>
  <messageID>1234567</messageID>
  ...
>>> s = str(mc)
>>> s
<mos>
  <mosID>MOS ID</mosID>
  <messageID>1234567</messageID>
  ...
```

2.3.3 Accessing the properties of a running order

For example, a *RunningOrder* object could contain several *Story* objects, each containing a number of *Item* objects:

```
>>> from mosromgr.mostypes import RunningOrder
>>> ro = RunningOrder.from_file('roCreate.mos.xml')
>>> ro.stories
[<Story 1234>, <Story 1235>, <Story 1236>]
>>> [story.duration for story in ro.stories]
[10, 20, 30]
>>> ro.duration
60
>>> story = ro.stories[0]
>>> story.slug
'Some story'
>>> story.items
[<Item ITEM1>, <Item ITEM2>, <Item ITEM3>]
>>> item = story.items[0]
>>> item.slug
'Some item'
```

In the case of a *StoryAppend* object, this would contain a single story:

```
>>> from mosromgr.mostypes import StoryAppend
>>> sa = StoryAppend.from_file('roStoryAppend.mos.xml')
>>> sa.story
```

(continues on next page)

(continued from previous page)

```
<Story STORY1>
>>> sa.duration
20
```

If this *StoryAppend* object was merged with a *RunningOrder* object, the new story would be accessible in the *RunningOrder* *stories* property:

```
>>> from mosromgr.mostypes import RunningOrder, StoryAppend
>>> ro = RunningOrder.from_file('roCreate.mos.xml')
>>> sa = StoryAppend.from_file('roStoryAppend.mos.xml')
>>> len(ro.stories)
3
>>> ro += sa
>>> len(ro.stories)
4
```

Additional information may be contained within the XML, and not exposed by properties the way *slug* and *object_id* are. In the spirit of *escape hatches* and *ejector seats*, the original XML in which the element was found is accessible as an `xml.etree.ElementTree.Element` object for further introspection.

For example, if you knew some of the `<item>` tags in a running order contained an `<areaID>` field, and you wanted to access its value, you could do so by inspecting the *xml* property:

```
tag = item.xml.find('areaID')
if tag is not None:
    print(tag.text)
```

2.3.4 Handling Exceptions

This can be useful for handling exceptions in your own code. For example, to handle any exception generated by the library, you can catch the library's base exception *MosRoMgrException*:

```
try:
    main()
except MosRoMgrException as e:
    print(e)
```

To catch a specific exception known to be raised under certain circumstances, each exception can be imported and handled separately if required:

```
from mosromgr.mostypes import MosFile
from mosromgr.exc import MosInvalidXML, UnknownMosFileType

try:
    ro = MosFile.from_file(mosfile)
except MosInvalidXML as e:
    print("Invalid in", mosfile)
except UnknownMosFileType as e:
    print("Unknown MOS file type", mosfile)
```

In some cases, a warning is raised rather than an exception. This means that execution is continued but a warning is output, which can be suppressed using the `warnings` module.

2.3.5 Capturing warnings

If you want to catch warnings and log them (for example, during a merge), you can use `warnings.catch_warnings`:

```
with warnings.catch_warnings(record=True) as warns:
    mc.merge()

warning_messages = [str(w.message) for w in warns]
```

2.3.6 Suppressing warnings

If you are not interested in seeing or capturing warnings, you can either use a `warning filter` or use `warnings.catch_warnings`:

```
with warnings.catch_warnings() as warns:
    mc.merge()
```

2.4 API

2.4.1 MOS Types

This part of the module provides the classes required for classifying and managing MOS files.

MOS Type classes are typically imported like so:

```
from mosromgr.mostypes import MosFile
```

MOS objects are constructed using one of three classmethods. Either from a file path:

```
ro = RunningOrder.from_file('roCreate.mos.xml')
```

from an XML string:

```
with open('roCreate.mos.xml') as f:
    xml = f.read()

ro = RunningOrder.from_string(xml)
```

or from an S3 file key:

```
ro = RunningOrder.from_s3(bucket_name='newsnight', mos_file_key='20200101/roCreate.mos.
↳ xml')
```

Similarly, objects constructed using these classmethods on the `MosFile` base class will be automatically classified and an instance of the relevant class will be created:

```
>>> ro = MosFile.from_file('roCreate.mos.xml')
>>> ro
<RunningOrder 1000>
>>> ss = MosFile.from_file('roStorySend.mos.xml')
>>> ss
<StorySend 1001>
>>> ro = MosFile.from_string(xml1)
>>> ro
<RunningOrder 1000>
>>> ss = MosFile.from_string(xml2)
>>> ss
<StorySend 1001>
```

Even roElementAction files, which require a number of different subclasses, can be classified this way:

```
>>> ea1 = MosFile.from_file('roElementAction1.mos.xml')
>>> ea1
<EAStorySwap 1012>
>>> ea2 = MosFile.from_string(xml)
>>> ea2
<EAItemMove 1013>
```

Note: Your AWS credentials must be configured to construct using the `from_s3()` classmethod. See <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

MOS message classes

The following classes are used to parse and manage specific types of MOS messages.

RunningOrder

class mosromgr.mostypes.RunningOrder

Bases: *mosromgr.mostypes.MosFile*

A RunningOrder object is created from a roCreate MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

Specification: Create Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-32

__add__(*other*)

RunningOrder objects can be merged with other MOS files which implement a merge method by using the + operator, for example:

```
ro = RunningOrder.from_file('roCreate.mos.xml')
ss = StorySend.from_file('roStorySend.mos.xml')
ro += ss
```

__gt__(*other*)

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(*other*)
Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()
The XML string of the MOS file

classmethod from_file(*mos_file_path*)
Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)
Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

property base_tag
The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property body
A list of elements found in the story bodies. Each item in the list is either a string (representing a <p> tag) or an *Item* object (representing an <item> tag). Unlike *script*, this does not exclude empty paragraph tags.

property completed
Whether or not the running order has had a *RunningOrderEnd* merged (*bool*)

property dict
Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property duration
Total running order duration in seconds (*int*)

property end_time
Transmission end time (*datetime.datetime*)

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property ro_slug
The running order slug (*str*)

property script
A list of strings found in paragraph tags within the story bodies, excluding any empty paragraphs or technical notes in brackets.

property start_time

Transmission start time (`datetime.datetime`) or `None` if not available in the XML

property stories

A list of [Story](#) objects within the running order

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

StorySend

class `mosromgr.mostypes.StorySend`

Bases: `mosromgr.mostypes.MosFile`

A `StorySend` object is created from a `roStorySend` MOS file and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

`StorySend` objects can be merged with a [RunningOrder](#) by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: Send Story information, including Body of the Story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-49

__gt__(other)

Sort by `message_id` i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by `message_id` i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod `from_file(mos_file_path)`

Construct from a path to a MOS file

Parameters `mos_file_path` (`str`) – The MOS file path

classmethod `from_s3(bucket_name, mos_file_key)`

Construct from a MOS file in an S3 bucket

Parameters

- `bucket_name` (`str`) – The name of the S3 bucket
- `mos_file_key` (`str`) – A MOS file key within the S3 bucket

classmethod `from_string(mos_xml_string)`

Construct from an XML string of a MOS document

Parameters `mos_xml_string` (`str`) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the [RunningOrder](#) object provided.

Replaces the story tag in the running order with the one in the `roStorySend` message.

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the `xml`, as determined by `base_tag_name`

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property story

The *Story* object being sent

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

StoryReplace**class mosromgr.mostypes.StoryReplace**

Bases: *mosromgr.mostypes.MosFile*

A StoryReplace object is created from a roStoryReplace MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

StoryReplace objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Replace a Story with Another in a Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roStoryReplace

__gt__(other)

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(mos_xml_string)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Replaces the story tag in the running order with the one in the *roStoryReplace* message.

property base_tag

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property stories

A list of replacement *Story* objects

property story

The *Story* object being replaced

property xml

The XML element of the MOS file (*xml.etree.ElementTree.Element*)

StoryInsert

class *mosromgr.mostypes.StoryInsert*

Bases: *mosromgr.mostypes.MosFile*

A *StoryInsert* object is created from a *roStoryInsert* MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

StoryInsert objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Insert Stories in Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roStoryInsert

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or *sorted([ro, ss])*

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or *sorted([ro, ss])*

__str__()

The XML string of the MOS file

classmethod *from_file(mos_file_path)*

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod *from_s3(bucket_name, mos_file_key)*

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket

- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters **mos_xml_string** (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(*ro*)
Merge into the *RunningOrder* object provided.
Inserts the story tags from the *roStoryInsert* message into the running order.

property base_tag
The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property dict
Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property source_stories
A list of *Story* objects to be inserted

property target_story
The *Story* object above which the source stories are to be inserted

property xml
The XML element of the MOS file (*xml.etree.ElementTree.Element*)

StoryAppend

class *mosromgr.mostypes.StoryAppend*

Bases: *mosromgr.mostypes.MosFile*

A *StoryAppend* object is created from a *roStoryAppend* MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

StoryAppend objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Append Stories to Running Order* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roStoryAppend

__gt__(*other*)
Sort by *message_id* i.e. *ss* > *ro* or *sorted*([*ro*, *ss*])

__lt__(*other*)
Sort by *message_id* i.e. *ro* < *ss* or *sorted*([*ro*, *ss*])

__str__()
The XML string of the MOS file

classmethod `from_file(mos_file_path)`

Construct from a path to a MOS file

Parameters `mos_file_path` (*str*) – The MOS file path

classmethod `from_s3(bucket_name, mos_file_key)`

Construct from a MOS file in an S3 bucket

Parameters

- `bucket_name` (*str*) – The name of the S3 bucket
- `mos_file_key` (*str*) – A MOS file key within the S3 bucket

classmethod `from_string(mos_xml_string)`

Construct from an XML string of a MOS document

Parameters `mos_xml_string` (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the *RunningOrder* object provided.

Adds the story tag in the `roStoryAppend` message onto the end of the running order.

property `base_tag`

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property `base_tag_name`

The name of the base XML tag for this file type (*str*)

property `dict`

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property `message_id`

The MOS file's message ID (*int*)

property `ro_id`

The running order ID (*str*)

property `stories`

A list of *Story* objects to be appended

property `xml`

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

StoryMove

class `mosromgr.mostypes.StoryMove`

Bases: `mosromgr.mostypes.MosFile`

A *StoryMove* object is created from a *roStoryMove* MOS file and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

StoryMove objects can be merged with a *RunningOrder* by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: Move a story to a new position in the Playlist http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roStoryMove

__gt__(*other*)
Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)
Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()
The XML string of the MOS file

classmethod from_file(*mos_file_path*)
Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)
Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(*ro*)
Merge into the *RunningOrder* object provided.
Moves the source story to the position above the target story.

property base_tag
The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property dict
Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property source_story
The *Story* object to be moved

property target_story
The *Story* object above which the source story is to be moved

property xml
The XML element of the MOS file (*xml.etree.ElementTree.Element*)

StoryDelete

class mosromgr.mostypes.StoryDelete

Bases: *mosromgr.mostypes.MosFile*

A StoryDelete object is created from a roStoryDelete MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

StoryDelete objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Delete Stories from Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roStoryDelete

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod **from_file**(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod **from_s3**(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod **from_string**(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Removes any story tags from the running order which are included in the roStoryDelete message.

property **base_tag**

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **dict**

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property **message_id**

The MOS file's message ID (*int*)

property **ro_id**

The running order ID (*str*)

property stories

A list of [Story](#) objects to be deleted

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

MetaDataReplace**class** mosromgr.mostypes.MetaDataReplace

Bases: [mosromgr.mostypes.MosFile](#)

A MetaDataReplace object is created from a roMetadataReplace MOS file and can be constructed using classmethods [from_file\(\)](#), [from_string\(\)](#) or [from_s3\(\)](#).

MetaDataReplace objects can be merged with a [RunningOrder](#) by using the + operator. This behaviour is defined in the [merge\(\)](#) method in this class.

Specification: Replace RO metadata without deleting the RO structure http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-34

__gt__(*other*)

Sort by [message_id](#) i.e. ss > ro or sorted([ro, ss])

__lt__(*other*)

Sort by [message_id](#) i.e. ro < ss or sorted([ro, ss])

__str__()

The XML string of the MOS file

classmethod [from_file](#)(*mos_file_path*)

Construct from a path to a MOS file

Parameters [mos_file_path](#) (*str*) – The MOS file path

classmethod [from_s3](#)(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- [bucket_name](#) (*str*) – The name of the S3 bucket
- [mos_file_key](#) (*str*) – A MOS file key within the S3 bucket

classmethod [from_string](#)(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters [mos_xml_string](#) (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the [RunningOrder](#) object provided.

Replaces the metadata tags in the running order with the ones in the MetaDataReplace message.

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by [base_tag_name](#)

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property ro_slug

The running order slug (*str*)

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

ItemDelete

class mosromgr.mostypes.ItemDelete

Bases: *mosromgr.mostypes.MosFile*

An ItemDelete object is created from a roItemDelete MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

ItemDelete objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Delete Items in Story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roItemDelete

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or *sorted([ro, ss])*

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or *sorted([ro, ss])*

__str__()

The XML string of the MOS file

classmethod from_file(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Deletes any item tags with the IDs specified in the `roItemDelete` message from the running order.

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the `xml`, as determined by `base_tag_name`

property base_tag_name

The name of the base XML tag for this file type (`str`)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (`dict`)

property items

A tuple of the two `Item` objects being deleted

property message_id

The MOS file's message ID (`int`)

property ro_id

The running order ID (`str`)

property story

The `Story` object containing the items being deleted

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

ItemInsert

class `mosromgr.mostypes.ItemInsert`

Bases: `mosromgr.mostypes.MosFile`

An `ItemInsert` object is created from a `roItemInsert` MOS file and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

`ItemInsert` objects can be merged with a `RunningOrder` by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: Insert Items in Story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roItemInsert

__gt__(*other*)

Sort by `message_id` i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(*other*)

Sort by `message_id` i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod `from_file(mos_file_path)`

Construct from a path to a MOS file

Parameters `mos_file_path` (`str`) – The MOS file path

classmethod `from_s3(bucket_name, mos_file_key)`

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (`str`) – The name of the S3 bucket
- **mos_file_key** (`str`) – A MOS file key within the S3 bucket

classmethod `from_string(mos_xml_string)`

Construct from an XML string of a MOS document

Parameters `mos_xml_string (str)` – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the *RunningOrder* object provided.

Inserts the item tags from the `roItemInsert` message into the relevant story in the running order.

property `base_tag`

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property `base_tag_name`

The name of the base XML tag for this file type (*str*)

property `dict`

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property `item`

The *Item* object above which the items are to be inserted

property `items`

A list of *Item* objects to be inserted

property `message_id`

The MOS file's message ID (*int*)

property `ro_id`

The running order ID (*str*)

property `story`

The *Story* object into which the items are to be inserted

property `xml`

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

ItemMoveMultiple

class `mosromgr.mostypes.ItemMoveMultiple`

Bases: `mosromgr.mostypes.MosFile`

An *ItemMoveMultiple* object is created from a *roItemMoveMultiple* MOS file and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

ItemMoveMultiple objects can be merged with a *RunningOrder* by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: Move one or more Items to a specified position within a Story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roItemMoveMultiple

`__gt__(other)`

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

`__lt__(other)`

Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

`__str__()`

The XML string of the MOS file

classmethod from_file(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Moves item tags in the *roItemMove* message to a new position within the story.

property base_tag

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property item

The *Item* object above which the items will be moved

property items

A list of *Item* objects to be moved

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property story

The *Story* object containing the items being moved

property xml

The XML element of the MOS file (*xml.etree.ElementTree.Element*)

ItemReplace

class mosromgr.mostypes.ItemReplace

Bases: *mosromgr.mostypes.MosFile*

An ItemReplace object is created from a roItemReplace MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

ItemReplace objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Replace an Item with one or more Items in a Story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOS_Protocol_Version_2.8.5_Final.htm#roItemReplace

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod **from_file**(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod **from_s3**(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod **from_string**(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Replaces the item tag in the story in the running order with the ones in the roItemReplace message

property **base_tag**

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **dict**

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property **item**

The *Item* object being replaced

property **items**

A list of replacement *Item* objects

property message_id
The MOS file's message ID (`int`)

property ro_id
The running order ID (`str`)

property story
The *Story* object containing the item being replaced

property xml
The XML element of the MOS file (`xml.etree.ElementTree.Element`)

ReadyToAir

class mosromgr.mostypes.ReadyToAir

Bases: *mosromgr.mostypes.MosFile*

A ReadyToAir object is created from a roReadyToAir MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

ReadyToAir objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Identify a Running Order as Ready to Air http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-41

__gt__(*other*)
Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(*other*)
Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()
The XML string of the MOS file

classmethod from_file(*mos_file_path*)
Construct from a path to a MOS file

Parameters *mos_file_path* (`str`) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)
Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (`str`) – The name of the S3 bucket
- **mos_file_key** (`str`) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters *mos_xml_string* (`str`) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(*ro*)
Merge into the *RunningOrder* object provided.
Currently unimplemented - has no effect on the running order. TODO: #18

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the `xml`, as determined by `base_tag_name`

property base_tag_name

The name of the base XML tag for this file type (`str`)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (`dict`)

property message_id

The MOS file's message ID (`int`)

property ro_id

The running order ID (`str`)

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EAStoryReplace

class mosromgr.mostypes.EAStoryReplace

Bases: `mosromgr.mostypes.ElementAction`

An EAStoryReplace object is created from a `roElementAction` MOS file containing a story replacement, and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

EAStoryReplace objects can be merged with a `RunningOrder` by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: Replacing a story http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(other)

Sort by `message_id` i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by `message_id` i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters `mos_file_path` (`str`) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- `bucket_name` (`str`) – The name of the S3 bucket
- `mos_file_key` (`str`) – A MOS file key within the S3 bucket

classmethod from_string(mos_xml_string)

Construct from an XML string of a MOS document

Parameters `mos_xml_string` (`str`) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the *RunningOrder* object provided.

Replaces the `element_target` story tag in the running order with any story tags found in the `element_source` in the `roElementAction` message.

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property stories

A list of replacement *Story* objects

property story

The *Story* object being replaced

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EItemReplace**class mosromgr.mostypes.EItemReplace**

Bases: *mosromgr.mostypes.ElementAction*

An *EItemReplace* object is created from a `roElementAction` MOS file containing an item replacement, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EItemReplace objects can be merged with a *RunningOrder* by using the `+` operator. This behaviour is defined in the *merge()* method in this class.

Specification: Replacing an item http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(other)

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters **mos_xml_string** (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Replaces the target item tag in the target story in the running order with any item tags found in the *element_source* in the *roElementAction* message.

property base_tag

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property item

The *Item* object being replaced

property items

A list of replacement *Item* objects

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property story

The *Story* object containing the item being replaced

property xml

The XML element of the MOS file (*xml.etree.ElementTree.Element*)

EAStoryDelete

class *mosromgr.mostypes.EAStoryDelete*

Bases: *mosromgr.mostypes.ElementAction*

An *EAStoryDelete* object is created from a *roElementAction* MOS file containing a story deletion, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EAStoryDelete objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Deleting stories* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or *sorted([ro, ss])*

__lt__(*other*)
Sort by *message_id* i.e. *ro* < *ss* or *sorted*(*[ro, ss]*)

__str__()
The XML string of the MOS file

classmethod from_file(*mos_file_path*)
Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)
Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(*ro*)
Merge into the *RunningOrder* object provided.

Removes any stories specified in *element_source* in the *roElementAction* message from the running order.

property base_tag
The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property dict
Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property stories
A list of *Story* objects to be deleted

property xml
The XML element of the MOS file (*xml.etree.ElementTree.Element*)

EItemDelete

class mosromgr.mostypes.EItemDelete

Bases: *mosromgr.mostypes.ElementAction*

An EItemDelete object is created from a roElementAction MOS file containing an item deletion, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EItemDelete objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Deleting items* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod **from_file**(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod **from_s3**(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod **from_string**(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Deletes any items specified in the *element_target* in the roStorySend message from the specified story in the running order.

property **base_tag**

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **dict**

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property **items**

A list of *Item* objects being deleted

property **message_id**

The MOS file's message ID (*int*)

property `ro_id`
 The running order ID (`str`)

property `story`
 The `Story` object containing the items being deleted

property `xml`
 The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EAStoryInsert

class `mosromgr.mostypes.EAStoryInsert`

Bases: `mosromgr.mostypes.ElementAction`

An `EAStoryInsert` object is created from a `roElementAction` MOS file containing a story insertion, and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

`EAStoryInsert` objects can be merged with a `RunningOrder` by using the `+` operator. This behaviour is defined in the `merge()` method in this class.

Specification: *Inserting stories* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

`__gt__(other)`
 Sort by `message_id` i.e. `ss > ro` or `sorted([ro, ss])`

`__lt__(other)`
 Sort by `message_id` i.e. `ro < ss` or `sorted([ro, ss])`

`__str__()`
 The XML string of the MOS file

classmethod `from_file(mos_file_path)`
 Construct from a path to a MOS file

Parameters `mos_file_path` (`str`) – The MOS file path

classmethod `from_s3(bucket_name, mos_file_key)`
 Construct from a MOS file in an S3 bucket

Parameters

- `bucket_name` (`str`) – The name of the S3 bucket
- `mos_file_key` (`str`) – A MOS file key within the S3 bucket

classmethod `from_string(mos_xml_string)`
 Construct from an XML string of a MOS document

Parameters `mos_xml_string` (`str`) – The XML string of the MOS document

inspect()
 Print an outline of the key file contents

merge(ro)
 Merge into the `RunningOrder` object provided.
 Inserts any story tags found in the `element_source` in the `roElementAction` message into the running order.

property `base_tag`
 The base tag (`xml.etree.ElementTree.Element`) within the `xml`, as determined by `base_tag_name`

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property stories

The *Story* objects to be inserted

property story

The *Story* object above which the source story will be inserted

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EAltemInsert

class mosromgr.mostypes.EAltemInsert

Bases: *mosromgr.mostypes.ElementAction*

An EAltemInsert object is created from a roElementAction MOS file containing an item insertion, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EAltemInsert objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Inserting items* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(other)

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(mos_xml_string)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(ro)
Merge into the *RunningOrder* object provided.

Inserts any item tags found in the `element_source` in the `roElementAction` message into the relevant story in the running order.

property base_tag
The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property dict
Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property item
The *Item* object above which the source item is to be inserted

property items
A list of *Item* objects to be inserted

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property story
The *Story* object into which the item is to be inserted

property xml
The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EAStorySwap

class `mosromgr.mostypes.EAStorySwap`

Bases: `mosromgr.mostypes.ElementAction`

An *EAStorySwap* object is created from a *roElementAction* MOS file containing a story swap, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EAStorySwap objects can be merged with a *RunningOrder* by using the `+` operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Swapping stories* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(other)
Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)
Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()
The XML string of the MOS file

classmethod from_file(mos_file_path)
Construct from a path to a MOS file

Parameters **mos_file_path** (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters **mos_xml_string** (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Swaps the order of the two story tags specified in *element_source* in the *roElementAction* message in the running order.

property base_tag

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property stories

A tuple of the two *Story* objects to be swapped

property xml

The XML element of the MOS file (*xml.etree.ElementTree.Element*)

EItemSwap

class mosromgr.mostypes.EItemSwap

Bases: *mosromgr.mostypes.ElementAction*

An EItemSwap object is created from a *roElementAction* MOS file containing an item swap, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EItemSwap objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Swapping items* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or *sorted([ro, ss])*

__lt__(*other*)
Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()
The XML string of the MOS file

classmethod from_file(*mos_file_path*)
Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)
Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)
Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()
Print an outline of the key file contents

merge(*ro*)
Merge into the *RunningOrder* object provided.

Swaps the order of the two item tags specified in *element_source* in the *roElementAction* message in the relevant story in the running order.

property base_tag
The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name
The name of the base XML tag for this file type (*str*)

property dict
Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property items
A tuple of the two *Item* objects to be swapped

property message_id
The MOS file's message ID (*int*)

property ro_id
The running order ID (*str*)

property story
The *Story* object containing the items being swapped

property xml
The XML element of the MOS file (*xml.etree.ElementTree.Element*)

EAStoryMove

class mosromgr.mostypes.EAStoryMove

Bases: *mosromgr.mostypes.ElementAction*

An EAStoryMove object is created from a *roElementAction* MOS file containing a story move, and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

EAStoryMove objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: *Moving stories* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod **from_file**(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod **from_s3**(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod **from_string**(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Moves story tags in *element_source* to the specified location in the running order.

property **base_tag**

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **dict**

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property **message_id**

The MOS file's message ID (*int*)

property **ro_id**

The running order ID (*str*)

property stories

A list of *Story* objects being moved

property story

The *Story* object above which the other stories will be moved

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

EItemMove**class mosromgr.mostypes.EItemMove**

Bases: `mosromgr.mostypes.ElementAction`

An EItemMove object is created from a `roElementAction` MOS file containing an item move, and can be constructed using classmethods `from_file()`, `from_string()` or `from_s3()`.

EItemMove objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the `merge()` method in this class.

Specification: *Moving items* http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-43

__gt__(other)

Sort by `message_id` i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by `message_id` i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters `mos_file_path` (*str*) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- `bucket_name` (*str*) – The name of the S3 bucket
- `mos_file_key` (*str*) – A MOS file key within the S3 bucket

classmethod from_string(mos_xml_string)

Construct from an XML string of a MOS document

Parameters `mos_xml_string` (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the *RunningOrder* object provided.

Moves item tags in `element_source` to the specified location in the story in the running order.

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by `base_tag_name`

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property item

The *Item* object above which the source items will be moved

property items

A list of *Item* objects to be moved

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property story

The *Story* object containing the item being replaced

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

RunningOrderReplace

class mosromgr.mostypes.RunningOrderReplace

Bases: *mosromgr.mostypes.RunningOrder*

An *RunningOrderReplace* object is created from a *roReplace* MOS file and can be constructed using class-methods *from_file()*, *from_string()* or *from_s3()*.

RunningOrderReplace objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class.

Specification: Replace Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-33

__add__(other)

RunningOrder objects can be merged with other MOS files which implement a merge method by using the + operator, for example:

```
ro = RunningOrder.from_file('roCreate.mos.xml')
ss = StorySend.from_file('roStorySend.mos.xml')
ro += ss
```

__gt__(other)

Sort by *message_id* i.e. `ss > ro` or `sorted([ro, ss])`

__lt__(other)

Sort by *message_id* i.e. `ro < ss` or `sorted([ro, ss])`

__str__()

The XML string of the MOS file

classmethod from_file(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod `from_s3(bucket_name, mos_file_key)`

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod `from_string(mos_xml_string)`

Construct from an XML string of a MOS document

Parameters **mos_xml_string** (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the [RunningOrder](#) object provided.

Replaces the entire roCreate tag in the running order with the one in the roReplace message.

property **base_tag**

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **body**

A list of elements found in the story bodies. Each item in the list is either a string (representing a <p> tag) or an *Item* object (representing an <item> tag). Unlike *script*, this does not exclude empty paragraph tags.

property **dict**

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property **duration**

Total running order duration in seconds (*int*)

property **end_time**

Transmission end time (`datetime.datetime`)

property **message_id**

The MOS file's message ID (*int*)

property **ro_id**

The running order ID (*str*)

property **ro_slug**

The running order slug (*str*)

property **script**

A list of strings found in paragraph tags within the story bodies, excluding any empty paragraphs or technical notes in brackets.

property **start_time**

Transmission start time (`datetime.datetime`) or `None` if not available in the XML

property **stories**

A list of *Story* objects within the running order

property **xml**

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

RunningOrderEnd

class mosromgr.mostypes.RunningOrderEnd

Bases: *mosromgr.mostypes.MosFile*

A RunningOrderEnd object is created from a roDelete MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

RunningOrderEnd objects can be merged with a *RunningOrder* by using the + operator. This behaviour is defined in the *merge()* method in this class. Once a RunningOrderEnd object has been merged into a *RunningOrder*, the running order is considered “completed” and no further messages can be merged (with the exception of *RunningOrderControl*).

Specification: Delete Running Order http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-35

__gt__(*other*)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(*other*)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod **from_file**(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod **from_s3**(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod **from_string**(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(*ro*)

Merge into the *RunningOrder* object provided.

Adds a mosromgrmeta tag containing the roDelete tag from the roDelete message to the roCreate tag in the running order.

property **base_tag**

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property **base_tag_name**

The name of the base XML tag for this file type (*str*)

property **dict**

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property **message_id**

The MOS file’s message ID (*int*)

property ro_id

The running order ID (*str*)

property xml

The XML element of the MOS file (*xml.etree.ElementTree.Element*)

RunningOrderControl**class mosromgr.mostypes.RunningOrderControl**

Bases: *mosromgr.mostypes.MosFile*

A RunningOrderControl object is created from a roCtrl MOS file and can be constructed using classmethods *from_file()*, *from_string()* or *from_s3()*.

Specification: Running Order Control http://mosprotocol.com/wp-content/MOS-Protocol-Documents/MOSProtocolVersion40/index.html#calibre_link-47

TODO: generalise this class #20

__gt__(other)

Sort by *message_id* i.e. *ss* > *ro* or sorted([*ro*, *ss*])

__lt__(other)

Sort by *message_id* i.e. *ro* < *ss* or sorted([*ro*, *ss*])

__str__()

The XML string of the MOS file

classmethod from_file(mos_file_path)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(bucket_name, mos_file_key)

Construct from a MOS file in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket
- **mos_file_key** (*str*) – A MOS file key within the S3 bucket

classmethod from_string(mos_xml_string)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

inspect()

Print an outline of the key file contents

merge(ro)

Merge into the *RunningOrder* object provided.

Replaces the story tag in the running order with the one in the *roStorySend* message

property base_tag

The base tag (*xml.etree.ElementTree.Element*) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using *xmltodict* library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (`int`)

property ro_id

The running order ID (`str`)

property story

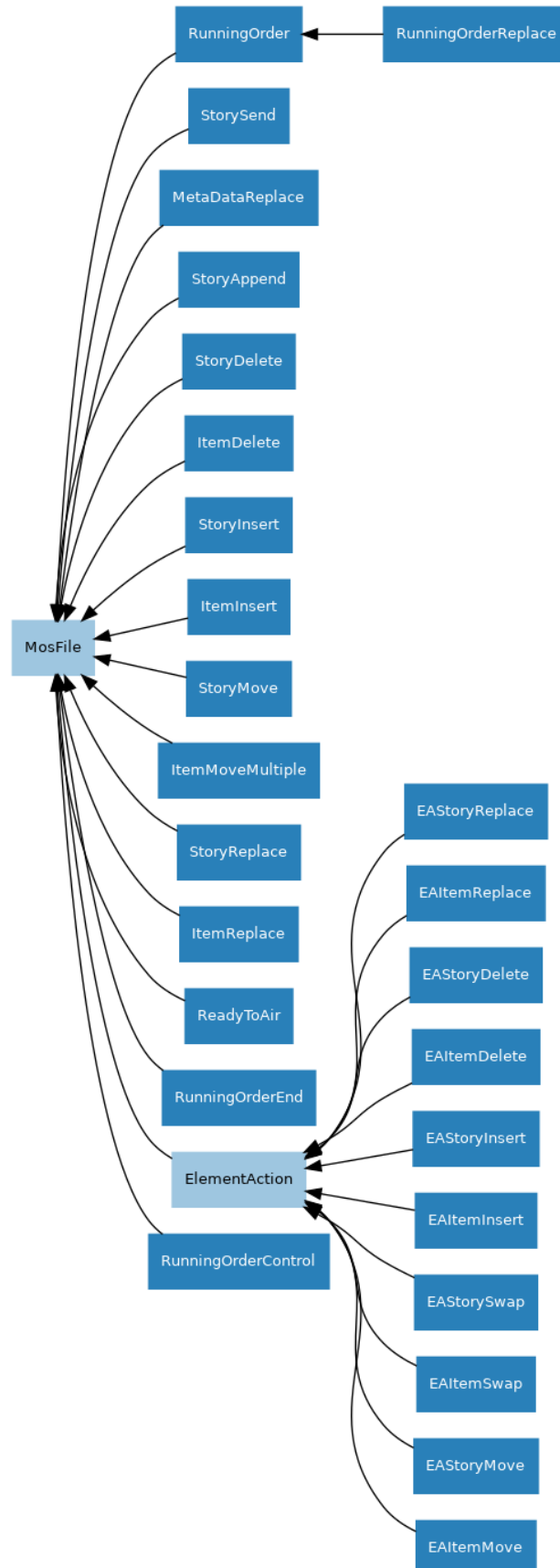
The story to which this roCtrl message relates

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

Base classes

Since some logic is shared between MOS file management, some inheritance is used in the implementation:



MosFile

class mosromgr.mostypes.MosFile

Base class for all MOS files

classmethod from_file(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters *mos_xml_string* (*str*) – The XML string of the MOS document

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

ElementAction

class mosromgr.mostypes.ElementAction

Base class for various roElementAction MOS files

classmethod from_file(*mos_file_path*)

Construct from a path to a MOS file

Parameters *mos_file_path* (*str*) – The MOS file path

classmethod from_s3(*bucket_name*, *mos_file_key*)

Construct from a MOS file in an S3 bucket

Parameters

- *bucket_name* (*str*) – The name of the S3 bucket
- *mos_file_key* (*str*) – A MOS file key within the S3 bucket

classmethod from_string(*mos_xml_string*)

Construct from an XML string of a MOS document

Parameters `mos_xml_string` (*str*) – The XML string of the MOS document

property base_tag

The base tag (`xml.etree.ElementTree.Element`) within the *xml*, as determined by *base_tag_name*

property base_tag_name

The name of the base XML tag for this file type (*str*)

property dict

Convert XML to dictionary using `xmltodict` library. Useful for testing. (*dict*)

property message_id

The MOS file's message ID (*int*)

property ro_id

The running order ID (*str*)

property xml

The XML element of the MOS file (`xml.etree.ElementTree.Element`)

2.4.2 MOS Elements

This part of the module provides a collection of classes used to provide easy access to certain elements within a *MosFile* object, such as a list of stories within a running order, and the items within a story.

Although usually not required directly, the MOS Element classes can be imported as follows:

```
from mosromgr.moselements import Story
```

Note: Note that these classes should not normally be constructed by the user, but instances of them can be found within *MosFile* objects, so the following documentation is provided as a reference to how they can be used.

Element classes

Story

class `mosromgr.moselements.Story`

Bases: `mosromgr.moselements.MosElement`

This class represents a Story element within any *MosFile* object, providing data relating to the story. The Story ID, Story slug, duration and more are exposed as properties, and the XML element is provided for further introspection.

__str__()

The XML string

property body

A list of elements found in the story body. Each item in the list is either a string (representing a `<p>` tag) or an *Item* object (representing an `<item>` tag). Unlike *script*, this does not exclude empty paragraph tags.

property duration

The story duration (the sum of the text time and media time found within `mosExternalMetadata->mosPayload`), in seconds (*float*)

property end_time

The transmission end time of the story (`datetime.datetime` or `None` if not available in the XML)

property id

The Story ID (`str`)

property items

List of `Item` elements found within the story (can be `None` if not available in the XML)

property offset

The time offset of the story in seconds (`float` or `None` if not available in the XML)

property script

A list of strings found in paragraph tags within the story body, excluding any empty paragraphs or technical notes in brackets.

property slug

The Story slug (`str` or `None` if not available in the XML)

property start_time

The transmission start time of the story (`datetime.datetime` or `None` if not available in the XML)

property xml

The XML element (`xml.etree.ElementTree.Element`)

Item

class `mosromgr.moselements.Item`

Bases: `mosromgr.moselements.MosElement`

This class represents an Item element within any `MosFile` object, providing data relating to an item within a `Story`. The Item ID and Item slug are exposed as properties, and the XML element is provided for further introspection.

__str__()

The XML string

property id

The Item ID (`str`)

property mos_id

The Item's MOS ID (`str` or `None` if not available in the XML)

property note

The item note text (`str` or `None` if not found)

property object_id

The Item's Object ID (`str` or `None` if not available in the XML)

property slug

The Item slug (`str` or `None` if not available in the XML)

property type

The Item's object type (`str` or `None` if not available in the XML)

property xml

The XML element (`xml.etree.ElementTree.Element`)

Base classes

MosElement

```
class mosromgr.moselements.MosElement
    Abstract base class for MOS elements

    __str__()
        The XML string

    property id
        The element ID (str)

    property slug
        The element slug (str or None if not available in the XML)

    property xml
        The XML element (xml.etree.ElementTree.Element)
```

2.4.3 MOS Collection

This part of the module provides a wrapper class *MosCollection* which stores references to specified MOS files, strings or S3 object keys so the *MosFile* objects can be recreated when needed rather than kept in memory.

Note: Note that creating a *MosCollection* from strings does not benefit from memory efficiency as the strings would still be held in memory.

The *MosCollection* is typically imported like so:

```
from mosromgr.moscollection import MosCollection
```

MOS collections are constructed using one of three classmethods. Either from a list of file paths:

```
mos_files = ['roCreate.mos.xml', 'roStorySend.mos.xml', 'roDelete.mos.xml']
mc = MosCollection.from_files(mos_files)
```

from a list of strings:

```
mos_strings = [roCreate, roStorySend, roDelete]
mc = MosCollection.from_strings(mos_files)
```

or from an S3 bucket:

```
mc = MosCollection.from_s3(bucket_name=bucket_name, prefix=prefix)
```

Note: Your AWS credentials must be configured to construct using the *from_s3()* classmethod. See <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

MosCollection

class mosromgr.moscollection.MosCollection

Wrapper for a collection of MOS files representing a partial or complete programme

`__str__()`

The XML string of the collection's running order

classmethod `from_files(mos_file_paths, *, allow_incomplete=False)`

Construct from a list of MOS file paths

Parameters

- **mos_file_paths** (*list*) – A list of paths to MOS files
- **allow_incomplete** (*bool*) – If False (the default), the collection is permitted to be constructed without a `roDelete`. If True, a [InvalidMosCollection](#) will be raised if one is not present. (keyword-only argument)

classmethod `from_s3(*, bucket_name, prefix, suffix='.mos.xml', allow_incomplete=False)`

Construct from a list of MOS files in an S3 bucket

Parameters

- **bucket_name** (*str*) – The name of the S3 bucket (keyword-only argument)
- **prefix** (*str*) – The prefix of the file keys in the S3 bucket (keyword-only argument)
- **suffix** (*str*) – The suffix of the file keys in the S3 bucket (keyword-only argument). Defaults to `'.mos.xml'`.
- **allow_incomplete** (*bool*) – If True, the collection is permitted to be constructed without a `roDelete`. If False (the default), a [InvalidMosCollection](#) will be raised if one is not present. (keyword-only argument)

classmethod `from_strings(mos_file_strings, *, allow_incomplete=False)`

Construct from a list of MOS document XML strings

Parameters

- **mos_file_paths** (*list*) – A list of paths to MOS files
- **allow_incomplete** (*bool*) – If False (the default), the collection is permitted to be constructed without a `roDelete`. If True, a [InvalidMosCollection](#) will be raised if one is not present. (keyword-only argument)

`merge(*, strict=True)`

Merge all MOS files into the collection's running order ([ro](#)). If *strict* is True (the default), then merge errors will be fatal. If False, then merge errors will be downgraded to warnings.

property `completed`

Whether or not the running order has had a [RunningOrderEnd](#) merged (*bool*)

property `mos_readers`

A list of [MosReader](#) objects representing all MOS files in the collection, except the [RunningOrder](#) (`roCreate`) which is held in [ro](#)

property `ro`

The collection's [RunningOrder](#) object

property `ro_id`

The running order ID

property ro_slug

The running order slug

MosReader

The *MosReader* class is internal and is not intended to be constructed by the user. A *MosCollection* object will contain a list of *MosReader* instances, so users may find it useful to refer to its members.

class mosromgr.moscollection.MosReader

Internal construct for opening and inspecting a MOS file for the purposes of classifying, sorting and validating a *MosCollection*. Provides the means to reconstruct the *MosFile* instance when needed in order to preserve memory usage.

property message_idThe message ID of the MOS file (*str*)**property mos_object**Restore the MOS object and return it (*MosFile*)**property mos_type**The *MosFile* subclass this object was classified as (returns the class object, not an instance or a string)**property ro_id**The MOS file's running order ID (*str*)**2.4.4 Utilities**

This part of the module provides a collection of generic utilities which are largely for internal use.

The various utilities are typically imported like so:

```
from mosromgr.utils import s3
```

Warning: This part of the module should not be considered part of the stable API and is subject to backwards-incompatible changes.

S3

AWS S3 utilities

get_mos_files

```
mosromgr.utils.s3.get_mos_files(bucket_name, prefix=None, *, suffix='.mos.xml')
```

Retrieve MOS files from given S3 bucket in location defined by *prefix*. Returns a list of file keys.

get_file_contents

`mosromgr.utils.s3.get_file_contents(bucket_name, file_key)`
Open the S3 file and return its contents as a string

XML

XML helper functions

remove_node

`mosromgr.utils.xml.remove_node(parent, node)`
Remove *node* from *parent*.

replace_node

`mosromgr.utils.xml.replace_node(parent, old_node, new_node, index)`
Replace *old_node* with *new_node* in *parent* at *index*.

insert_node

`mosromgr.utils.xml.insert_node(parent, node, index)`
Insert *node* in *parent* at *index*.

find_child

`mosromgr.utils.xml.find_child(parent, child_tag, id=None)`
Find an element with *child_tag* in *parent* and return (*child*, *index*) or (*None*, *None*) if not found. If *id* is provided, it will be searched for, otherwise the first child will be returned.

2.4.5 Exceptions

The module's exceptions and warnings are typically imported like so:

```
from mosromgr.exc import MosRoMgrException
```

The library's base warning is *MosRoMgrWarning* and others are detailed below.

Errors

MosRoMgrException

exception `mosromgr.exc.MosRoMgrException`

Bases: `Exception`

Base class for all mosromgr exceptions

UnknownMosFileType

exception `mosromgr.exc.UnknownMosFileType`

Bases: `mosromgr.exc.MosRoMgrException`

Exception raised when a MOS file type cannot be determined

MosMergeError

exception `mosromgr.exc.MosMergeError`

Bases: `mosromgr.exc.MosRoMgrException`

Exception raised when MOS merge fails

MosCompletedMergeError

exception `mosromgr.exc.MosCompletedMergeError`

Bases: `mosromgr.exc.MosMergeError`

Exception raised when MOS merge is attempted on a completed *RunningOrder*

InvalidMosCollection

exception `mosromgr.exc.InvalidMosCollection`

Bases: `mosromgr.exc.MosRoMgrException`

Exception raised when MosCollection fails validation

MosInvalidXML

exception `mosromgr.exc.MosInvalidXML`

Bases: `mosromgr.exc.MosRoMgrException`

Exception raised when *MosFile* cannot parse given XML

Warnings

MosRoMgrWarning

exception `mosromgr.exc.MosRoMgrWarning`

Bases: `Warning`

Base class for all warnings in mosromgr

MosMergeNonStrictWarning

exception `mosromgr.exc.MosMergeNonStrictWarning`

Bases: `mosromgr.exc.MosRoMgrWarning`

Warning raised when a merge error occurs in non-strict mode

ItemNotFoundWarning

exception `mosromgr.exc.ItemNotFoundWarning`

Bases: `mosromgr.exc.MosRoMgrWarning`

Warning raised when an item cannot be found during a *MosFile* merge

StoryNotFoundWarning

exception `mosromgr.exc.StoryNotFoundWarning`

Bases: `mosromgr.exc.MosRoMgrWarning`

Warning raised when a story cannot be found during a *MosFile* merge

DuplicateStoryWarning

exception `mosromgr.exc.DuplicateStoryWarning`

Bases: `mosromgr.exc.MosRoMgrWarning`

Warning raised when a story being added is already found during a *EAStoryInsert* merge

2.5 CLI

This section provides a reference to the command line interface.

2.5.1 mosromgr detect

Detect the MOS type of one or more files

Synopsis

```
mosromgr detect [-h] [-f [files [files ...]]] [-b bucket] [-p prefix] [-s suffix] [-k ↵key]
```

Description

-f --files [files ...]
The MOS files to detect

-b --bucket-name bucket
The name of the S3 bucket containing the MOS files

-p --prefix prefix
The prefix for MOS files in the S3 bucket

-s --suffix suffix
The suffix for MOS files in the S3 bucket

-k --key key
The file key for a MOS file in the S3 bucket

-h, --help
Show this help message and exit

Usage

Detect the type of a MOS file:

```
$ mosromgr detect -f 123456-roCreate.mos.xml
123456-roCreate.mos.xml: RunningOrder
```

Multiple files can be provided as arguments:

```
$ mosromgr detect -f 123456-roCreate.mos.xml 123457-roStorySend.mos.xml
123456-roCreate.mos.xml: RunningOrder
123457-roStorySend.mos.xml: StorySend
```

Wildcards can also be used:

```
$ mosromgr detect *
123456-roCreate.mos.xml: RunningOrder
123457-roStorySend.mos.xml: StorySend
...
9148627-roDelete.mos.xml: RunningOrderEnd
bbcProgrammeMetadata.xml: Unknown MOS file type
cricket: Invalid
FINAL.json: Invalid
FINAL.xml: RunningOrder (completed)
```

You can also read files from an S3 bucket. Either a specific file by key:

```
$ mosromgr detect -b my-bucket -k newscnight/20210101/123456-roCreate.mos.xml
OPENMEDIA_NCS.W1.BBC.MOS/OM_10.1253459/5744992-roCreate.mos.xml: RunningOrder
```

Or a whole folder by prefix:

```
$ mosromgr detect -b bbc-news-labs-slicer-mos-message-store -p newscnight/20210101/
newscnight/20210101/123456-roCreate.mos.xml: RunningOrder
newscnight/20210101/123457-roStorySend.mos.xml: StorySend
newscnight/20210101/123458-roStorySend.mos.xml: StorySend
```

(continues on next page)

(continued from previous page)

```
newsnight/20210101/123459-roStorySend.mos.xml: StorySend
...
```

Note: Your AWS credentials must be configured to use the S3 method. See <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

2.5.2 mosromgr inspect

View the high-level the contents of a MOS file

Synopsis

```
mosromgr inspect [-h] [-f [files [files ...]]] [-b bucket] [-p prefix] [-s suffix] [-k ↵
↵key]
```

Description

- f --files [files ...]**
The MOS files to inspect
- b --bucket-name bucket**
The name of the S3 bucket containing the MOS files
- p --prefix prefix**
The prefix for MOS files in the S3 bucket
- s --suffix suffix**
The suffix for MOS files in the S3 bucket
- k --key key**
The file key for a MOS file in the S3 bucket
- h, --help**
Show this help message and exit

Usage

View the contents of a local MOS file:

```
$ mosromgr inspect -f 123456-roCreate.mos.xml
RO: 22:45 NEWSNIGHT 54D CORE Thu, 08.04.2021
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529414,4.15529413.1
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529416,4.15529413.3
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529418,4.15529413.5
...
```

View the contents of a MOS file in S3:

```
$ mosromgr inspect -b my-bucket -k newsnight/20210804/123456-roCreate.mos.xml
RO: 22:45 NEWSNIGHT 54D CORE Thu, 08.04.2021
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529414,4.15529413.1
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529416,4.15529413.3
STORY: OPENMEDIA_NCS.W1.BBC.MOS;OM_4.15529413;OM_4.15529418,4.15529413.5
```

Note: Your AWS credentials must be configured to use the S3 method. See <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

2.5.3 mosromgr merge

Merge the provided MOS files

Synopsis

```
mosromgr merge [-h] [-f [files [files ...]]] [-b bucket] [-p prefix] [-s suffix]
                [-o outfile] [-i] [-n]
```

Description

- f --files [files ...]**
The MOS files to merge
- b --bucket-name bucket**
The name of the S3 bucket containing the MOS files
- p --prefix prefix**
The file prefix for MOS files in the S3 bucket
- s --suffix suffix**
The file suffix for MOS files in the S3 bucket
- o --outfile outfile**
Output to a file
- i --incomplete**
Allow an incomplete collection
- n --non-strict**
Downgrade MOS merge errors to warnings
- h, --help**
Show this help message and exit

Usage

Merge local files and store the result in a new file:

```
$ mosromgr merge -f *.mos.xml -o FINAL.xml
...
INFO:mosromgr.moscollection:Merging RunningOrderEnd 123499
INFO:mosromgr.moscollection:Completed merging 99 mos files
Writing merged running order to FINAL.xml
```

Merge files in an S3 bucket folder by prefix and store the result in a new file:

```
$ mosromgr merge -b my-bucket -p newnight/20210101/ -o FINAL.xml
...
INFO:mosromgr.moscollection:Merging RunningOrderEnd 123499
INFO:mosromgr.moscollection:Completed merging 99 mos files
Writing merged running order to FINAL.xml
```

Note: Your AWS credentials must be configured to use the S3 method. See <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/credentials.html>

2.6 Uses of mosromgr

This section lists projects which have been known to use the *mosromgr* module. If you have used *mosromgr* in a project and would like to add it to the list, please either [edit this file](#) and open a pull request, [open an issue](#), or send an email to bbcnewslabsteam@bbc.co.uk.

2.6.1 BBC News Labs - MOS pipeline

We have a collection of [AWS](#) services making up a pipeline which processes MOS messages in real time, updates a status dashboard, publishes completed MOS running orders and JSON summaries to an internal document store, and populates a directory of programmes with new episodes and lists of stories (complete with timing information) as they become available.

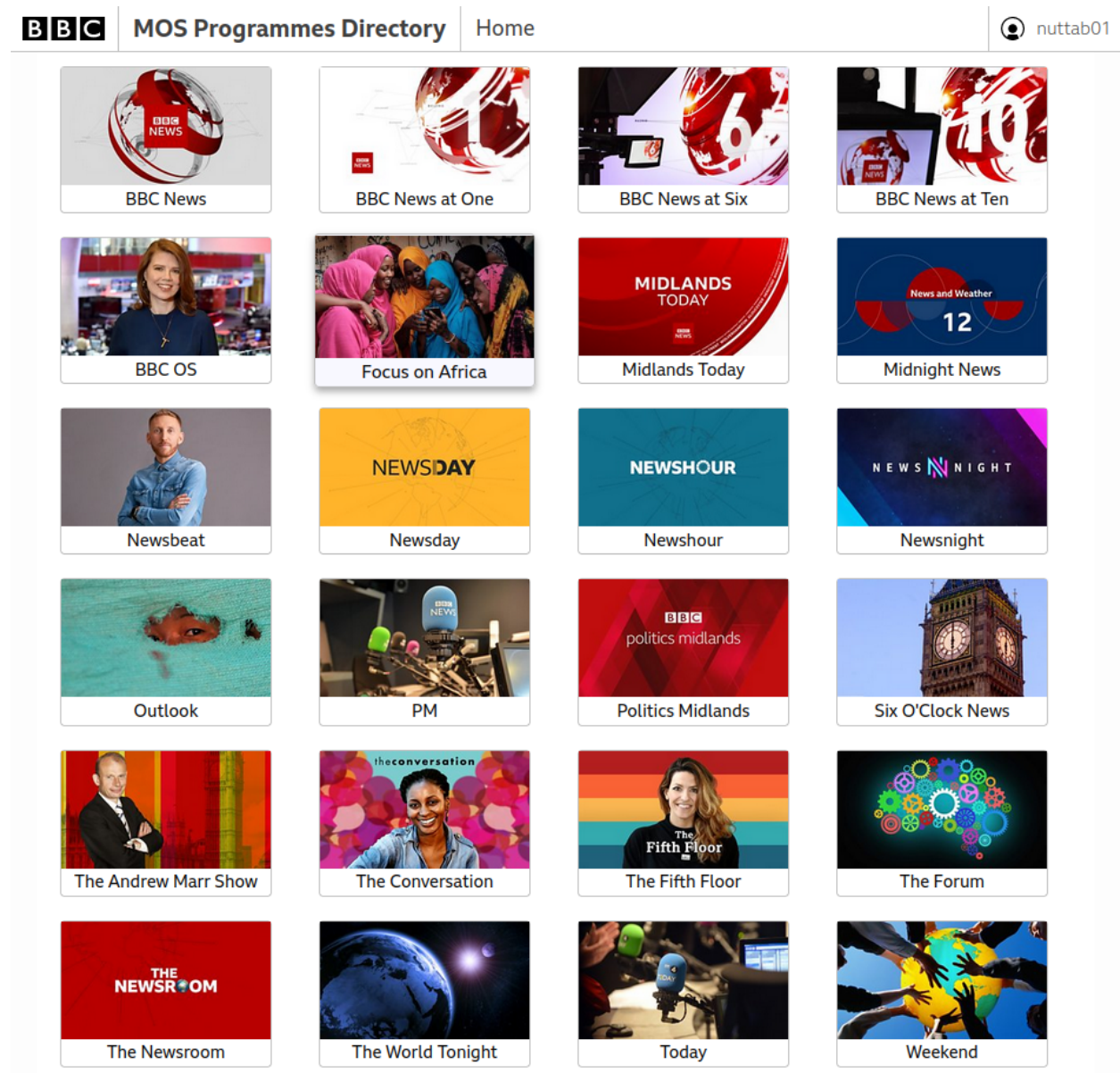
Status dashboard:

[Home](#) | [Status page](#)

Running order status page

jobs		completed (4,981)		pending (58)		error (0)		
RO ID	RO Slug	Files	Brand Title	Episode Title	First Seen	Last Seen	Warnings	
OPENMEDIA_NCS.BIRMINGHAM.BBC.MOS.OM_2.7484062	1830 MIDLANDS TODAY Tue, 22.06.2021	3,307	Midlands Today	Evening News, 22/06/2021	2021-06-22 12:43	2021-06-28 16:46		
OPENMEDIA_NCS.W1.BBC.MOS.OM_3.14004728	1400 W5 Newshour Mon, 28.06.2021	343	Newshour	28/06/2021 13:06 GMT	2021-06-28 11:03	2021-06-28 15:47	► 6 warnings	
OPENMEDIA_NCS.BIRMINGHAM.BBC.MOS.OM_2.7487862	1330 MIDLANDS TODAY Mon, 28.06.2021	411	Midlands Today	Lunchtime News, 28/06/2021	2021-06-28 09:59	2021-06-28 15:46		
OPENMEDIA_NCS.SALFORD.BBC.MOS.OM_3.11293812	You and Yours Mon, 28.06.2021	143	You and Yours	28/06/2021	2021-06-28 08:58	2021-06-28 15:46		
OPENMEDIA_NCS.W1.BBC.MOS.OM_3.14004390	1600 Focus on Africa Mon, 28.06.2021	127	Focus on Africa	28/06/2021 GMT	2021-06-28 14:12	2021-06-28 15:46		
OPENMEDIA_NCS.W1.BBC.MOS.OM_7.3875140	Newsbeat 1245 Mon, 28.06.2021	98	Newsbeat	28/06/2021	2021-06-28 10:17	2021-06-28 15:46	► 1 warnings	
OPENMEDIA_NCS.W1.BBC.MOS.OM_4.15315625	1300 BBC NEWS AT ONE Mon, 28.06.2021	661	BBC News at One	28/06/2021	2021-06-28 11:10	2021-06-28 15:42		
OPENMEDIA_NCS.W1.BBC.MOS.OM_10.2136100	34D 1200B ARABIC TV NEWS Mon, 28.06.2021	784			2021-06-28 10:53	2021-06-28 15:41		
OPENMEDIA_NCS.W1.BBC.MOS.OM_10.2137391	54D 1330 PERSIAN NEWS Mon, 28.06.2021	636			2021-06-28 11:53	2021-06-28 15:41		
OPENMEDIA_NCS.W1.BBC.MOS.OM_10.2151656	54D 1200 PERSIAN REITH PILOT Mon, 28.06.2021	556			2021-06-28 09:34	2021-06-28 15:41		
OPENMEDIA_NCS.W1.BBC.MOS.OM_4.15315726	1200 NEWS CHANNEL + BBC Two Mon, 28.06.2021	580			2021-06-28 10:04	2021-06-28 15:41		
OPENMEDIA_NCS.W1.BBC.MOS.OM_5.609053	1300 R4 WATO Mon, 28.06.2021	597	World at One	28/06/2021	2021-06-28 09:19	2021-06-28 15:41		
OPENMEDIA_NCS.W1.BBC.MOS.OM_4.15316007	1600 NEWS CHANNEL Mon, 28.06.2021	574			2021-06-28 14:03	2021-06-28 15:40		
OPENMEDIA_NCS.W1.BBC.MOS.OM_4.15315820	1400 NEWS CHANNEL Mon, 28.06.2021	408			2021-06-28 12:35	2021-06-28 15:40		
OPENMEDIA_NCS.W1.BBC.MOS.OM_10.2136179	34D 1600B TRENDING Mon, 28.06.2021	363			2021-06-28 14:10	2021-06-28 15:40		
OPENMEDIA_NCS.W1.BBC.MOS.OM_10.2136875	34D 1400B WORLD AT ONE ARABIC TV NEWS Mon, 28.06.2021	364			2021-06-28 12:26	2021-06-28 15:40		
OPENMEDIA_NCS.W1.BBC.MOS.OM_4.15315913	1500 NEWS CHANNEL Mon, 28.06.2021	413			2021-06-28 13:06	2021-06-28 15:40		
OPENMEDIA_NCS.W1.BBC.MOS.OM_3.14004795	1500 W5 Newshour Mon, 28.06.2021	351			2021-06-28 11:03	2021-06-28 15:40	► 4 warnings	

Programmes directory:




Example chapterised breakdown of an episode of [Newsnight](#):

BBC
MOS Programmes Directory
Newsnight: 16/06/2021
nuttat01

Home » Newsnight » 16/06/2021 » script

Newsnight: 16/06/2021



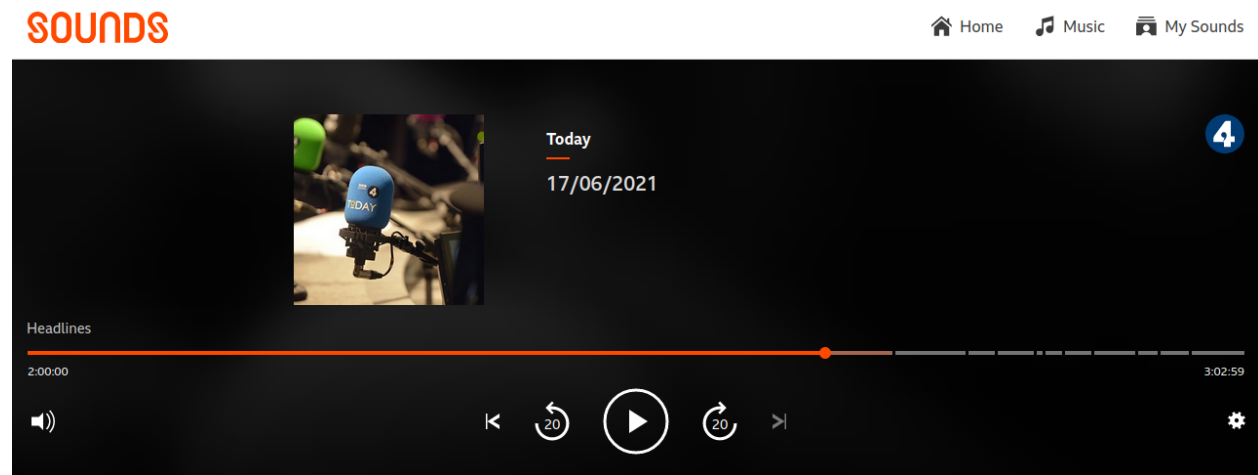
The day's important national and international news stories. With Emily Maitlis.

Stories in this episode

1	PLAY	MENU	0:00:00	2021-06-16T21:45:00+01:00
2	PLAY	BIDEN/PUTIN	0:01:05	2021-06-16T21:46:05+01:00
3	PLAY	CUMMINGS	0:13:27	2021-06-16T21:58:27+01:00
4	PLAY	POLICE	0:15:38	2021-06-16T22:00:38+01:00
5	PLAY	ASYLUM	0:26:18	2021-06-16T22:11:18+01:00
6	PLAY	CUMMINGS REPRISE	0:31:31	2021-06-16T22:16:31+01:00
7	PLAY	PAPERS	0:36:56	2021-06-16T22:21:56+01:00
8	PLAY	GOODBYE	0:37:56	2021-06-16T22:22:56+01:00

2.6.2 BBC News Labs - Auto chapterisation

We were able to decorate the player timeline with chapter points in certain BBC TV and radio programmes:




We used the script and story timing information extracted from the running order and aligned it against the transcript.

Released On: 17 Jun 2021 Available for 29 days


News and current affairs, including Sports Desk, Weather and Thought for the Day.

 Subscribe

 Bookmark

 More episodes

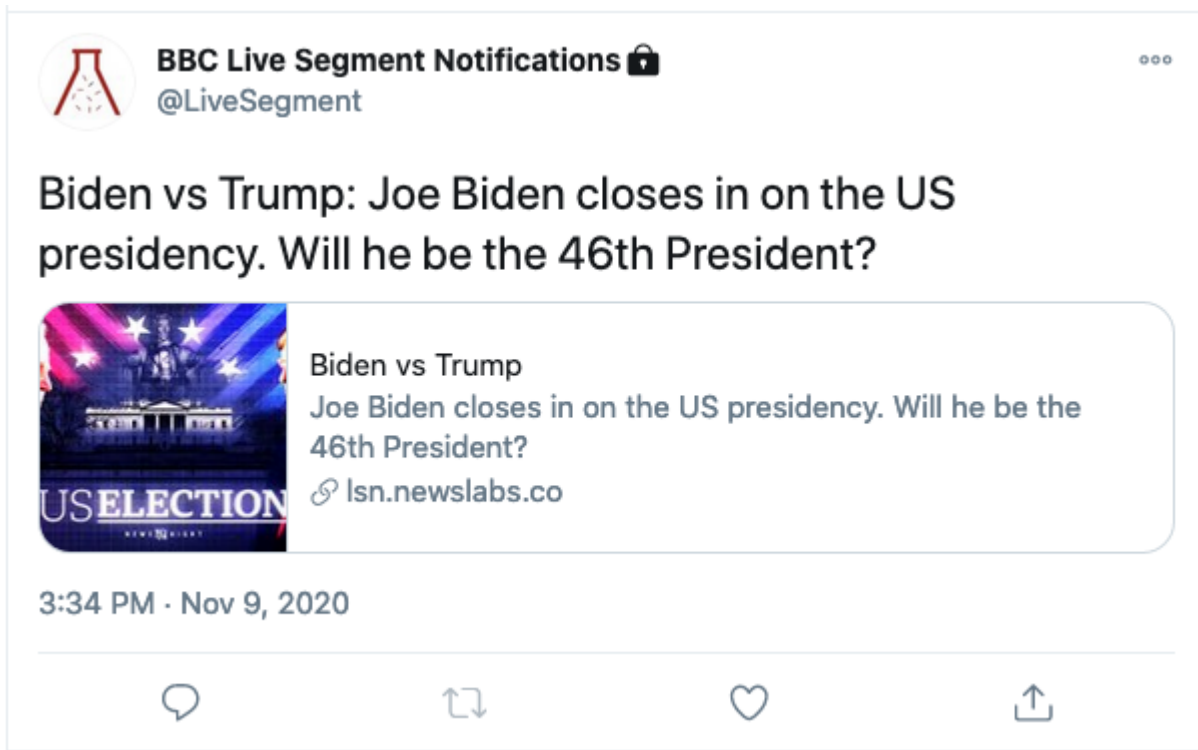
[Programme Website](#) 

 Chapters in this episode ▲

▶ Headlines	120:00
Headlines and news bulletins at eight o'clock on Thursday 17 June	
▶ Is help for business enough?	130:01
Is help for business enough?: Treasury Minister Jesse Norman on how business will be affected by 21 June delay	
▶ Influencer rule-breakers to be named	141:00
The ASA is launching a website page naming online influencers who break advertising rules	
▶ Sport news	145:32
Sport news with Karthi Gnanasegaram	
▶ News summary	151:20
Weather and news summary at half past eight	
▶ UK-Australia trade deal	152:39
The government publishes negotiation details for a UK-Australia free trade deal	
▶ Lockdown Tory rebels	155:35
Over 50 Conservatives voted against England's four week extension of Covid restrictions	
▶ Older people impacted by economy	160:02
IFS report shows Covid's impact on the economy could affect the eldest more than the young	
▶ Tigray humanitarian crisis	166:35
Tigray, Ethiopia, faces a humanitarian crisis following seven months of war	
▶ Children's social care 'under significant strain'	169:53
A reports finds children's social care in England is failing to provide support	
▶ Picture books	174:40
Author Julia Donaldson shares her secret on how to write for the very youngest readers	

2.6.3 BBC News Labs - Live Segment Notifications

We developed a proof-of-concept in which a note within a story in a running order could trigger a tweet to alert people of an upcoming story in time to watch live, or link to the clip of the story on-demand:



2.7 Changelog

Warning: Note that the library is currently in beta. The API and CLI are not yet stable and may change. Once the library reaches v1.0, it will be considered stable. Please consider giving [Feedback](#) to help stabilise the API.

2.7.1 Release 0.9.1 (2021-09-02)

- Add `type`, `object_id`, and `mos_id` properties to the `Item` class

2.7.2 Release 0.9.0 (2021-06-21)

- Updated `mosromgr inspect` CLI command to work for all file types
- Corrected some singular `MosFile MOS element` properties that should have been lists (e.g. `source_story` should have been `source_stories`)
- Improved validation and error handling when merging various `MosFile` objects
- Added `script` and `body` to `Story`
- Added `script` and `body` to `RunningOrder`
- Added non-strict mode to the `MosCollection merge()` method and the `mosromgr merge` CLI command

- Corrected some edge cases in *MosFile* subclass merge implementations (e.g. empty storyID tag means move to bottom)

2.7.3 Release 0.8.1 (2021-04-14)

- Fixup release

2.7.4 Release 0.8.0 (2021-04-13)

- Improved validation and error handling when merging various *MosFile* objects
- Added more arguments to CLI commands
- Corrected some singular *MosFile MOS Elements* properties that should have been lists (e.g. `source_story` should have been `source_stories`)

2.7.5 Release 0.7.0 (2021-01-08)

- Ensured exceptions are raised when story IDs are not found when merging
- Ensured tags aren't overwritten when they are empty in *MetaDataReplace*
- Ensured target story is found when merging *StoryInsert* and *StoryReplace*
- Added *RunningOrderControl* class (for `roCtrl` messages)
- Changed `tx_time` to `start_time`

2.7.6 Release 0.6.0 (2020-12-01)

- Added support for `<StoryDuration>` as an alternative to `<MediaTime>` and `<TextTime>`

2.7.7 Release 0.5.0 (2020-11-30)

- Added *ReadyToAir* MOS Type
- Improved error message on invalid *MosCollection*

2.7.8 Release 0.4.0 (2020-11-30)

- Changed `closed` property to `completed`
- Added transmission time and offset to *Story* class
- New *CLI* with separate commands for *mosromgr detect*, *mosromgr inspect* and *mosromgr merge*
- Make *MosCollection* raise exceptions on failure, not just warnings

2.7.9 Release 0.3.0 (2020-11-24)

- Switched from complicated `__init__` constructors to multiple `from_` classmethods e.g. `from_file()`
- Replaced `get_mos_object` function with detection logic in the `MosFile` and `ElementAction` base classes
- Replaced `MosContainer` class with `MosCollection`

2.7.10 Release 0.2.0 (2020-11-24)

- Added `MOS Elements` - a collection of classes used to provide easy access to certain elements within a `MosFile` object

2.7.11 Release 0.1.0 (2020-11-24)

- Implemented most standard MOS message types as `MosFile` subclasses, supporting merging subsequent messages into the original running order
- Implemented a MOS file detection function (`get_mos_object`)
- Added a `MosContainer` class as a wrapper for a complete programme
- Added a `CLI` for merging MOS files

2.8 Development

This page contains reference material for those interested in developing and contributing to the **mosromgr** module. The project source code is hosted on GitHub at <https://github.com/bbc/mosromgr> which also includes the [issue tracker](#).

2.8.1 Setting up for Development

1. Clone the repository and enter the directory:

```
$ git clone https://github.com/bbc/mosromgr
$ cd mosromgr
```

2. Create a virtual environment e.g. using `virtualenvwrapper`:

```
$ mkvirtualenv mosromgr
```

3. Install the project for development:

```
$ make develop
```

After completing these steps, the library and command line interface will be available to use within your environment. Any modifications made to the source code will be automatically reflected within the environment.

2.8.2 Tests

The test suite uses [pytest](#). Tests are organised mirroring the source code.

Running the tests

To run the linter, test suite and coverage analysis, activate the environment and run:

```
$ make test
```

For more control when running tests, run `pytest` directly, for example `pytest -vvvk story` will run tests with `story` in the name (`-k story`) with verbose output (`-vv`), and stop at the first failure (`-x`).

2.8.3 Documentation

The documentation is built using [sphinx](#) using the [diataxis](#) framework.

Building the documentation

To build the documentation, activate the environment and run:

```
$ make doc
```

This will generate the required diagrams and build the HTML docs which will be located in `docs/build/html`. Serve them with the command:

```
$ make doc-serve
```

You'll now be able to open the docs on your browser at `http://localhost:8000/`.

2.9 Feedback

Before we release v1.0 and stabilise the API, we are seeking other organisations using the MOS protocol to test *mosromgr* on their own MOS files and provide feedback so we can integrate any necessary changes to make sure it works effectively beyond the BBC's use.

If you can help, please test the module on your own MOS files and report back to us using our [discussion board](#) or [issue tracker](#) on GitHub, or email us at bbcnewslabsteam@bbc.co.uk.

2.10 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

ISSUES AND QUESTIONS

Questions can be asked on the [discussion board](#), and issues can be raised on the [issue tracker](#).

CONTRIBUTING

Source code can be found on GitHub at github.com/bbc/mosromgr.

Contributions are welcome. Please refer to the [contributing guidelines](#).

CONTRIBUTORS

- Ben Nuttall
- Owen Turlamain
- Rob French
- Lucy MacGlashan
- Dave Bevan
- Matthew Sim

LICENCE

Licensed under the [Apache License, Version 2.0](#).

CONTACT

To get in touch with the maintainers, please contact the BBC News Labs team: bbcnewslabsteam@bbc.co.uk



PYTHON MODULE INDEX

m

- `mosromgr.exc`, [52](#)
- `mosromgr.moscollection`, [49](#)
- `mosromgr.moselements`, [47](#)
- `mosromgr.mostypes`, [11](#)
- `mosromgr.utils`, [51](#)

Symbols

- `__add__()` (*mosromgr.mostypes.RunningOrder* method), 12
- `__add__()` (*mosromgr.mostypes.RunningOrderReplace* method), 40
- `__gt__()` (*mosromgr.mostypes.EAItemDelete* method), 32
- `__gt__()` (*mosromgr.mostypes.EAItemInsert* method), 34
- `__gt__()` (*mosromgr.mostypes.EAItemMove* method), 39
- `__gt__()` (*mosromgr.mostypes.EAItemReplace* method), 29
- `__gt__()` (*mosromgr.mostypes.EAItemSwap* method), 36
- `__gt__()` (*mosromgr.mostypes.EAStoryDelete* method), 30
- `__gt__()` (*mosromgr.mostypes.EAStoryInsert* method), 33
- `__gt__()` (*mosromgr.mostypes.EAStoryMove* method), 38
- `__gt__()` (*mosromgr.mostypes.EAStoryReplace* method), 28
- `__gt__()` (*mosromgr.mostypes.EAStorySwap* method), 35
- `__gt__()` (*mosromgr.mostypes.ItemDelete* method), 22
- `__gt__()` (*mosromgr.mostypes.ItemInsert* method), 23
- `__gt__()` (*mosromgr.mostypes.ItemMoveMultiple* method), 24
- `__gt__()` (*mosromgr.mostypes.ItemReplace* method), 26
- `__gt__()` (*mosromgr.mostypes.MetaDataReplace* method), 21
- `__gt__()` (*mosromgr.mostypes.ReadyToAir* method), 27
- `__gt__()` (*mosromgr.mostypes.RunningOrder* method), 12
- `__gt__()` (*mosromgr.mostypes.RunningOrderControl* method), 43
- `__gt__()` (*mosromgr.mostypes.RunningOrderEnd* method), 42
- `__gt__()` (*mosromgr.mostypes.RunningOrderReplace* method), 40
- `__gt__()` (*mosromgr.mostypes.StoryAppend* method), 17
- `__gt__()` (*mosromgr.mostypes.StoryDelete* method), 20
- `__gt__()` (*mosromgr.mostypes.StoryInsert* method), 16
- `__gt__()` (*mosromgr.mostypes.StoryMove* method), 18
- `__gt__()` (*mosromgr.mostypes.StoryReplace* method), 15
- `__gt__()` (*mosromgr.mostypes.StorySend* method), 14
- `__lt__()` (*mosromgr.mostypes.EAItemDelete* method), 32
- `__lt__()` (*mosromgr.mostypes.EAItemInsert* method), 34
- `__lt__()` (*mosromgr.mostypes.EAItemMove* method), 39
- `__lt__()` (*mosromgr.mostypes.EAItemReplace* method), 29
- `__lt__()` (*mosromgr.mostypes.EAItemSwap* method), 36
- `__lt__()` (*mosromgr.mostypes.EAStoryDelete* method), 30
- `__lt__()` (*mosromgr.mostypes.EAStoryInsert* method), 33
- `__lt__()` (*mosromgr.mostypes.EAStoryMove* method), 38
- `__lt__()` (*mosromgr.mostypes.EAStoryReplace* method), 28
- `__lt__()` (*mosromgr.mostypes.EAStorySwap* method), 35
- `__lt__()` (*mosromgr.mostypes.ItemDelete* method), 22
- `__lt__()` (*mosromgr.mostypes.ItemInsert* method), 23
- `__lt__()` (*mosromgr.mostypes.ItemMoveMultiple* method), 24
- `__lt__()` (*mosromgr.mostypes.ItemReplace* method), 26
- `__lt__()` (*mosromgr.mostypes.MetaDataReplace* method), 21
- `__lt__()` (*mosromgr.mostypes.ReadyToAir* method), 27
- `__lt__()` (*mosromgr.mostypes.RunningOrder* method), 12
- `__lt__()` (*mosromgr.mostypes.RunningOrderControl* method), 43
- `__lt__()` (*mosromgr.mostypes.RunningOrderEnd* method), 42
- `__lt__()` (*mosromgr.mostypes.RunningOrderReplace* method), 40
- `__lt__()` (*mosromgr.mostypes.StoryAppend* method), 17
- `__lt__()` (*mosromgr.mostypes.StoryDelete* method), 20
- `__lt__()` (*mosromgr.mostypes.StoryInsert* method), 16
- `__lt__()` (*mosromgr.mostypes.StoryMove* method), 19

`__lt__()` (*mosromgr.mostypes.StoryReplace method*), 15
`__lt__()` (*mosromgr.mostypes.StorySend method*), 14
`__str__()` (*mosromgr.moscollection.MosCollection method*), 50
`__str__()` (*mosromgr.moselements.Item method*), 48
`__str__()` (*mosromgr.moselements.MosElement method*), 49
`__str__()` (*mosromgr.moselements.Story method*), 47
`__str__()` (*mosromgr.mostypes.EAItemDelete method*), 32
`__str__()` (*mosromgr.mostypes.EAItemInsert method*), 34
`__str__()` (*mosromgr.mostypes.EAItemMove method*), 39
`__str__()` (*mosromgr.mostypes.EAItemReplace method*), 29
`__str__()` (*mosromgr.mostypes.EAItemSwap method*), 37
`__str__()` (*mosromgr.mostypes.EAStoryDelete method*), 31
`__str__()` (*mosromgr.mostypes.EAStoryInsert method*), 33
`__str__()` (*mosromgr.mostypes.EAStoryMove method*), 38
`__str__()` (*mosromgr.mostypes.EAStoryReplace method*), 28
`__str__()` (*mosromgr.mostypes.EAStorySwap method*), 35
`__str__()` (*mosromgr.mostypes.ItemDelete method*), 22
`__str__()` (*mosromgr.mostypes.ItemInsert method*), 23
`__str__()` (*mosromgr.mostypes.ItemMoveMultiple method*), 24
`__str__()` (*mosromgr.mostypes.ItemReplace method*), 26
`__str__()` (*mosromgr.mostypes.MetadataReplace method*), 21
`__str__()` (*mosromgr.mostypes.ReadyToAir method*), 27
`__str__()` (*mosromgr.mostypes.RunningOrder method*), 13
`__str__()` (*mosromgr.mostypes.RunningOrderControl method*), 43
`__str__()` (*mosromgr.mostypes.RunningOrderEnd method*), 42
`__str__()` (*mosromgr.mostypes.RunningOrderReplace method*), 40
`__str__()` (*mosromgr.mostypes.StoryAppend method*), 17
`__str__()` (*mosromgr.mostypes.StoryDelete method*), 20
`__str__()` (*mosromgr.mostypes.StoryInsert method*), 16
`__str__()` (*mosromgr.mostypes.StoryMove method*), 19
`__str__()` (*mosromgr.mostypes.StoryReplace method*), 15
`__str__()` (*mosromgr.mostypes.StorySend method*), 14
`--help`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57
`-b --bucket-name bucket`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57
`-f --files [files ...]`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57
`-h`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57
`-i --incomplete`
 mosromgr-merge command line option, 57
`-k --key key`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
`-n --non-strict`
 mosromgr-merge command line option, 57
`-o --outfile outfile`
 mosromgr-merge command line option, 57
`-p --prefix prefix`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57
`-s --suffix suffix`
 mosromgr-detect command line option, 55
 mosromgr-inspect command line option, 56
 mosromgr-merge command line option, 57

B

`base_tag` (*mosromgr.mostypes.EAItemDelete property*), 32
`base_tag` (*mosromgr.mostypes.EAItemInsert property*), 35
`base_tag` (*mosromgr.mostypes.EAItemMove property*), 39
`base_tag` (*mosromgr.mostypes.EAItemReplace property*), 30
`base_tag` (*mosromgr.mostypes.EAItemSwap property*), 37
`base_tag` (*mosromgr.mostypes.EAStoryDelete property*), 31
`base_tag` (*mosromgr.mostypes.EAStoryInsert property*), 33
`base_tag` (*mosromgr.mostypes.EAStoryMove property*), 38
`base_tag` (*mosromgr.mostypes.EAStoryReplace property*), 29

`base_tag` (`mosromgr.mostypes.EAStorySwap` property), 36
`base_tag` (`mosromgr.mostypes.ElementAction` property), 47
`base_tag` (`mosromgr.mostypes.ItemDelete` property), 23
`base_tag` (`mosromgr.mostypes.ItemInsert` property), 24
`base_tag` (`mosromgr.mostypes.ItemMoveMultiple` property), 25
`base_tag` (`mosromgr.mostypes.ItemReplace` property), 26
`base_tag` (`mosromgr.mostypes.MetaDataReplace` property), 21
`base_tag` (`mosromgr.mostypes.MosFile` property), 46
`base_tag` (`mosromgr.mostypes.ReadyToAir` property), 27
`base_tag` (`mosromgr.mostypes.RunningOrder` property), 13
`base_tag` (`mosromgr.mostypes.RunningOrderControl` property), 43
`base_tag` (`mosromgr.mostypes.RunningOrderEnd` property), 42
`base_tag` (`mosromgr.mostypes.RunningOrderReplace` property), 41
`base_tag` (`mosromgr.mostypes.StoryAppend` property), 18
`base_tag` (`mosromgr.mostypes.StoryDelete` property), 20
`base_tag` (`mosromgr.mostypes.StoryInsert` property), 17
`base_tag` (`mosromgr.mostypes.StoryMove` property), 19
`base_tag` (`mosromgr.mostypes.StoryReplace` property), 16
`base_tag` (`mosromgr.mostypes.StorySend` property), 14
`base_tag_name` (`mosromgr.mostypes.EAItemDelete` property), 32
`base_tag_name` (`mosromgr.mostypes.EAItemInsert` property), 35
`base_tag_name` (`mosromgr.mostypes.EAItemMove` property), 39
`base_tag_name` (`mosromgr.mostypes.EAItemReplace` property), 30
`base_tag_name` (`mosromgr.mostypes.EAItemSwap` property), 37
`base_tag_name` (`mosromgr.mostypes.EAStoryDelete` property), 31
`base_tag_name` (`mosromgr.mostypes.EAStoryInsert` property), 33
`base_tag_name` (`mosromgr.mostypes.EAStoryMove` property), 38
`base_tag_name` (`mosromgr.mostypes.EAStoryReplace` property), 29
`base_tag_name` (`mosromgr.mostypes.EAStorySwap` property), 36
`base_tag_name` (`mosromgr.mostypes.ElementAction` property), 47
`base_tag_name` (`mosromgr.mostypes.ItemDelete` property), 23
`base_tag_name` (`mosromgr.mostypes.ItemInsert` property), 24
`base_tag_name` (`mosromgr.mostypes.ItemMoveMultiple` property), 25
`base_tag_name` (`mosromgr.mostypes.ItemReplace` property), 26
`base_tag_name` (`mosromgr.mostypes.MetaDataReplace` property), 21
`base_tag_name` (`mosromgr.mostypes.MosFile` property), 46
`base_tag_name` (`mosromgr.mostypes.ReadyToAir` property), 28
`base_tag_name` (`mosromgr.mostypes.RunningOrder` property), 13
`base_tag_name` (`mosromgr.mostypes.RunningOrderControl` property), 43
`base_tag_name` (`mosromgr.mostypes.RunningOrderEnd` property), 42
`base_tag_name` (`mosromgr.mostypes.RunningOrderReplace` property), 41
`base_tag_name` (`mosromgr.mostypes.StoryAppend` property), 18
`base_tag_name` (`mosromgr.mostypes.StoryDelete` property), 20
`base_tag_name` (`mosromgr.mostypes.StoryInsert` property), 17
`base_tag_name` (`mosromgr.mostypes.StoryMove` property), 19
`base_tag_name` (`mosromgr.mostypes.StoryReplace` property), 16
`base_tag_name` (`mosromgr.mostypes.StorySend` property), 14
`body` (`mosromgr.moselements.Story` property), 47
`body` (`mosromgr.mostypes.RunningOrder` property), 13
`body` (`mosromgr.mostypes.RunningOrderReplace` property), 41

C

`completed` (`mosromgr.moscollection.MosCollection` property), 50
`completed` (`mosromgr.mostypes.RunningOrder` property), 13

D

`dict` (`mosromgr.mostypes.EAItemDelete` property), 32
`dict` (`mosromgr.mostypes.EAItemInsert` property), 35
`dict` (`mosromgr.mostypes.EAItemMove` property), 40
`dict` (`mosromgr.mostypes.EAItemReplace` property), 30
`dict` (`mosromgr.mostypes.EAItemSwap` property), 37
`dict` (`mosromgr.mostypes.EAStoryDelete` property), 31
`dict` (`mosromgr.mostypes.EAStoryInsert` property), 34
`dict` (`mosromgr.mostypes.EAStoryMove` property), 38
`dict` (`mosromgr.mostypes.EAStoryReplace` property), 29
`dict` (`mosromgr.mostypes.EAStorySwap` property), 36

`dict (mosromgr.mostypes.ElementAction property)`, 47
`dict (mosromgr.mostypes.ItemDelete property)`, 23
`dict (mosromgr.mostypes.ItemInsert property)`, 24
`dict (mosromgr.mostypes.ItemMoveMultiple property)`, 25
`dict (mosromgr.mostypes.ItemReplace property)`, 26
`dict (mosromgr.mostypes.MetaDataReplace property)`, 21
`dict (mosromgr.mostypes.MosFile property)`, 46
`dict (mosromgr.mostypes.ReadyToAir property)`, 28
`dict (mosromgr.mostypes.RunningOrder property)`, 13
`dict (mosromgr.mostypes.RunningOrderControl property)`, 43
`dict (mosromgr.mostypes.RunningOrderEnd property)`, 42
`dict (mosromgr.mostypes.RunningOrderReplace property)`, 41
`dict (mosromgr.mostypes.StoryAppend property)`, 18
`dict (mosromgr.mostypes.StoryDelete property)`, 20
`dict (mosromgr.mostypes.StoryInsert property)`, 17
`dict (mosromgr.mostypes.StoryMove property)`, 19
`dict (mosromgr.mostypes.StoryReplace property)`, 16
`dict (mosromgr.mostypes.StorySend property)`, 15
`DuplicateStoryWarning`, 54
`duration (mosromgr.moselements.Story property)`, 47
`duration (mosromgr.mostypes.RunningOrder property)`, 13
`duration (mosromgr.mostypes.RunningOrderReplace property)`, 41

E

`EItemDelete (class in mosromgr.mostypes)`, 32
`EItemInsert (class in mosromgr.mostypes)`, 34
`EItemMove (class in mosromgr.mostypes)`, 39
`EItemReplace (class in mosromgr.mostypes)`, 29
`EItemSwap (class in mosromgr.mostypes)`, 36
`EStoryDelete (class in mosromgr.mostypes)`, 30
`EStoryInsert (class in mosromgr.mostypes)`, 33
`EStoryMove (class in mosromgr.mostypes)`, 38
`EStoryReplace (class in mosromgr.mostypes)`, 28
`EStorySwap (class in mosromgr.mostypes)`, 35
`ElementAction (class in mosromgr.mostypes)`, 46
`end_time (mosromgr.moselements.Story property)`, 47
`end_time (mosromgr.mostypes.RunningOrder property)`, 13
`end_time (mosromgr.mostypes.RunningOrderReplace property)`, 41

F

`find_child() (in module mosromgr.utils.xml)`, 52
`from_file() (mosromgr.mostypes.EItemDelete class method)`, 32
`from_file() (mosromgr.mostypes.EItemInsert class method)`, 34

`from_file() (mosromgr.mostypes.EItemMove class method)`, 39
`from_file() (mosromgr.mostypes.EItemReplace class method)`, 29
`from_file() (mosromgr.mostypes.EItemSwap class method)`, 37
`from_file() (mosromgr.mostypes.EStoryDelete class method)`, 31
`from_file() (mosromgr.mostypes.EStoryInsert class method)`, 33
`from_file() (mosromgr.mostypes.EStoryMove class method)`, 38
`from_file() (mosromgr.mostypes.EStoryReplace class method)`, 28
`from_file() (mosromgr.mostypes.EStorySwap class method)`, 35
`from_file() (mosromgr.mostypes.ElementAction class method)`, 46
`from_file() (mosromgr.mostypes.ItemDelete class method)`, 22
`from_file() (mosromgr.mostypes.ItemInsert class method)`, 23
`from_file() (mosromgr.mostypes.ItemMoveMultiple class method)`, 24
`from_file() (mosromgr.mostypes.ItemReplace class method)`, 26
`from_file() (mosromgr.mostypes.MetaDataReplace class method)`, 21
`from_file() (mosromgr.mostypes.MosFile class method)`, 46
`from_file() (mosromgr.mostypes.ReadyToAir class method)`, 27
`from_file() (mosromgr.mostypes.RunningOrder class method)`, 13
`from_file() (mosromgr.mostypes.RunningOrderControl class method)`, 43
`from_file() (mosromgr.mostypes.RunningOrderEnd class method)`, 42
`from_file() (mosromgr.mostypes.RunningOrderReplace class method)`, 40
`from_file() (mosromgr.mostypes.StoryAppend class method)`, 17
`from_file() (mosromgr.mostypes.StoryDelete class method)`, 20
`from_file() (mosromgr.mostypes.StoryInsert class method)`, 16
`from_file() (mosromgr.mostypes.StoryMove class method)`, 19
`from_file() (mosromgr.mostypes.StoryReplace class method)`, 15
`from_file() (mosromgr.mostypes.StorySend class method)`, 14
`from_files() (mosromgr.moscollection.MosCollection class method)`, 50

<code>from_s3()</code> (<i>mosromgr.moscollection.MosCollection</i> class method), 50	<code>from_s3()</code> (<i>mosromgr.mostypes.StoryReplace</i> class method), 15
<code>from_s3()</code> (<i>mosromgr.mostypes.EAItemDelete</i> class method), 32	<code>from_s3()</code> (<i>mosromgr.mostypes.StorySend</i> class method), 14
<code>from_s3()</code> (<i>mosromgr.mostypes.EAItemInsert</i> class method), 34	<code>from_string()</code> (<i>mosromgr.mostypes.EAItemDelete</i> class method), 32
<code>from_s3()</code> (<i>mosromgr.mostypes.EAItemMove</i> class method), 39	<code>from_string()</code> (<i>mosromgr.mostypes.EAItemInsert</i> class method), 34
<code>from_s3()</code> (<i>mosromgr.mostypes.EAItemReplace</i> class method), 29	<code>from_string()</code> (<i>mosromgr.mostypes.EAItemMove</i> class method), 39
<code>from_s3()</code> (<i>mosromgr.mostypes.EAItemSwap</i> class method), 37	<code>from_string()</code> (<i>mosromgr.mostypes.EAItemReplace</i> class method), 30
<code>from_s3()</code> (<i>mosromgr.mostypes.EAStoryDelete</i> class method), 31	<code>from_string()</code> (<i>mosromgr.mostypes.EAItemSwap</i> class method), 37
<code>from_s3()</code> (<i>mosromgr.mostypes.EAStoryInsert</i> class method), 33	<code>from_string()</code> (<i>mosromgr.mostypes.EAStoryDelete</i> class method), 31
<code>from_s3()</code> (<i>mosromgr.mostypes.EAStoryMove</i> class method), 38	<code>from_string()</code> (<i>mosromgr.mostypes.EAStoryInsert</i> class method), 33
<code>from_s3()</code> (<i>mosromgr.mostypes.EAStoryReplace</i> class method), 28	<code>from_string()</code> (<i>mosromgr.mostypes.EAStoryMove</i> class method), 38
<code>from_s3()</code> (<i>mosromgr.mostypes.EAStorySwap</i> class method), 36	<code>from_string()</code> (<i>mosromgr.mostypes.EAStoryReplace</i> class method), 28
<code>from_s3()</code> (<i>mosromgr.mostypes.ElementAction</i> class method), 46	<code>from_string()</code> (<i>mosromgr.mostypes.EAStorySwap</i> class method), 36
<code>from_s3()</code> (<i>mosromgr.mostypes.ItemDelete</i> class method), 22	<code>from_string()</code> (<i>mosromgr.mostypes.ElementAction</i> class method), 46
<code>from_s3()</code> (<i>mosromgr.mostypes.ItemInsert</i> class method), 23	<code>from_string()</code> (<i>mosromgr.mostypes.ItemDelete</i> class method), 22
<code>from_s3()</code> (<i>mosromgr.mostypes.ItemMoveMultiple</i> class method), 25	<code>from_string()</code> (<i>mosromgr.mostypes.ItemInsert</i> class method), 23
<code>from_s3()</code> (<i>mosromgr.mostypes.ItemReplace</i> class method), 26	<code>from_string()</code> (<i>mosromgr.mostypes.ItemMoveMultiple</i> class method), 25
<code>from_s3()</code> (<i>mosromgr.mostypes.MetadataReplace</i> class method), 21	<code>from_string()</code> (<i>mosromgr.mostypes.ItemReplace</i> class method), 26
<code>from_s3()</code> (<i>mosromgr.mostypes.MosFile</i> class method), 46	<code>from_string()</code> (<i>mosromgr.mostypes.MetadataReplace</i> class method), 21
<code>from_s3()</code> (<i>mosromgr.mostypes.ReadyToAir</i> class method), 27	<code>from_string()</code> (<i>mosromgr.mostypes.MosFile</i> class method), 46
<code>from_s3()</code> (<i>mosromgr.mostypes.RunningOrder</i> class method), 13	<code>from_string()</code> (<i>mosromgr.mostypes.ReadyToAir</i> class method), 27
<code>from_s3()</code> (<i>mosromgr.mostypes.RunningOrderControl</i> class method), 43	<code>from_string()</code> (<i>mosromgr.mostypes.RunningOrder</i> class method), 13
<code>from_s3()</code> (<i>mosromgr.mostypes.RunningOrderEnd</i> class method), 42	<code>from_string()</code> (<i>mosromgr.mostypes.RunningOrderControl</i> class method), 43
<code>from_s3()</code> (<i>mosromgr.mostypes.RunningOrderReplace</i> class method), 40	<code>from_string()</code> (<i>mosromgr.mostypes.RunningOrderEnd</i> class method), 42
<code>from_s3()</code> (<i>mosromgr.mostypes.StoryAppend</i> class method), 18	<code>from_string()</code> (<i>mosromgr.mostypes.RunningOrderReplace</i> class method), 41
<code>from_s3()</code> (<i>mosromgr.mostypes.StoryDelete</i> class method), 20	<code>from_string()</code> (<i>mosromgr.mostypes.StoryAppend</i> class method), 18
<code>from_s3()</code> (<i>mosromgr.mostypes.StoryInsert</i> class method), 16	<code>from_string()</code> (<i>mosromgr.mostypes.StoryDelete</i> class method), 20
<code>from_s3()</code> (<i>mosromgr.mostypes.StoryMove</i> class method), 19	<code>from_string()</code> (<i>mosromgr.mostypes.StoryInsert</i> class method), 17

`from_string()` (*mosromgr.mostypes.StoryMove* class method), 19
`from_string()` (*mosromgr.mostypes.StoryReplace* class method), 15
`from_string()` (*mosromgr.mostypes.StorySend* class method), 14
`from_strings()` (*mosromgr.moscollection.MosCollection* class method), 50

G

`get_file_contents()` (in module *mosromgr.utils.s3*), 52
`get_mos_files()` (in module *mosromgr.utils.s3*), 51

I

`id` (*mosromgr.moselements.Item* property), 48
`id` (*mosromgr.moselements.MosElement* property), 49
`id` (*mosromgr.moselements.Story* property), 47
`insert_node()` (in module *mosromgr.utils.xml*), 52
`inspect()` (*mosromgr.mostypes.EAItemDelete* method), 32
`inspect()` (*mosromgr.mostypes.EAItemInsert* method), 34
`inspect()` (*mosromgr.mostypes.EAItemMove* method), 39
`inspect()` (*mosromgr.mostypes.EAItemReplace* method), 30
`inspect()` (*mosromgr.mostypes.EAItemSwap* method), 37
`inspect()` (*mosromgr.mostypes.EAStoryDelete* method), 31
`inspect()` (*mosromgr.mostypes.EAStoryInsert* method), 33
`inspect()` (*mosromgr.mostypes.EAStoryMove* method), 38
`inspect()` (*mosromgr.mostypes.EAStoryReplace* method), 28
`inspect()` (*mosromgr.mostypes.EAStorySwap* method), 36
`inspect()` (*mosromgr.mostypes.ItemDelete* method), 22
`inspect()` (*mosromgr.mostypes.ItemInsert* method), 24
`inspect()` (*mosromgr.mostypes.ItemMoveMultiple* method), 25
`inspect()` (*mosromgr.mostypes.ItemReplace* method), 26
`inspect()` (*mosromgr.mostypes.MetadataReplace* method), 21
`inspect()` (*mosromgr.mostypes.ReadyToAir* method), 27
`inspect()` (*mosromgr.mostypes.RunningOrder* method), 13
`inspect()` (*mosromgr.mostypes.RunningOrderControl* method), 43
`inspect()` (*mosromgr.mostypes.RunningOrderEnd* method), 42
`inspect()` (*mosromgr.mostypes.RunningOrderReplace* method), 41
`inspect()` (*mosromgr.mostypes.StoryAppend* method), 18
`inspect()` (*mosromgr.mostypes.StoryDelete* method), 20
`inspect()` (*mosromgr.mostypes.StoryInsert* method), 17
`inspect()` (*mosromgr.mostypes.StoryMove* method), 19
`inspect()` (*mosromgr.mostypes.StoryReplace* method), 15
`inspect()` (*mosromgr.mostypes.StorySend* method), 14
`InvalidMosCollection`, 53
`Item` (class in *mosromgr.moselements*), 48
`item` (*mosromgr.mostypes.EAItemInsert* property), 35
`item` (*mosromgr.mostypes.EAItemMove* property), 40
`item` (*mosromgr.mostypes.EAItemReplace* property), 30
`item` (*mosromgr.mostypes.ItemInsert* property), 24
`item` (*mosromgr.mostypes.ItemMoveMultiple* property), 25
`item` (*mosromgr.mostypes.ItemReplace* property), 26
`ItemDelete` (class in *mosromgr.mostypes*), 22
`ItemInsert` (class in *mosromgr.mostypes*), 23
`ItemMoveMultiple` (class in *mosromgr.mostypes*), 24
`ItemNotFoundWarning`, 54
`ItemReplace` (class in *mosromgr.mostypes*), 26
`items` (*mosromgr.moselements.Story* property), 48
`items` (*mosromgr.mostypes.EAItemDelete* property), 32
`items` (*mosromgr.mostypes.EAItemInsert* property), 35
`items` (*mosromgr.mostypes.EAItemMove* property), 40
`items` (*mosromgr.mostypes.EAItemReplace* property), 30
`items` (*mosromgr.mostypes.EAItemSwap* property), 37
`items` (*mosromgr.mostypes.ItemDelete* property), 23
`items` (*mosromgr.mostypes.ItemInsert* property), 24
`items` (*mosromgr.mostypes.ItemMoveMultiple* property), 25
`items` (*mosromgr.mostypes.ItemReplace* property), 26

M

`merge()` (*mosromgr.moscollection.MosCollection* method), 50
`merge()` (*mosromgr.mostypes.EAItemDelete* method), 32
`merge()` (*mosromgr.mostypes.EAItemInsert* method), 35
`merge()` (*mosromgr.mostypes.EAItemMove* method), 39
`merge()` (*mosromgr.mostypes.EAItemReplace* method), 30
`merge()` (*mosromgr.mostypes.EAItemSwap* method), 37
`merge()` (*mosromgr.mostypes.EAStoryDelete* method), 31
`merge()` (*mosromgr.mostypes.EAStoryInsert* method), 33
`merge()` (*mosromgr.mostypes.EAStoryMove* method), 38
`merge()` (*mosromgr.mostypes.EAStoryReplace* method), 28
`merge()` (*mosromgr.mostypes.EAStorySwap* method), 36

[merge\(\)](#) (*mosromgr.mostypes.ItemDelete* method), 22
[merge\(\)](#) (*mosromgr.mostypes.ItemInsert* method), 24
[merge\(\)](#) (*mosromgr.mostypes.ItemMoveMultiple* method), 25
[merge\(\)](#) (*mosromgr.mostypes.ItemReplace* method), 26
[merge\(\)](#) (*mosromgr.mostypes.MetadataReplace* method), 21
[merge\(\)](#) (*mosromgr.mostypes.ReadyToAir* method), 27
[merge\(\)](#) (*mosromgr.mostypes.RunningOrderControl* method), 43
[merge\(\)](#) (*mosromgr.mostypes.RunningOrderEnd* method), 42
[merge\(\)](#) (*mosromgr.mostypes.RunningOrderReplace* method), 41
[merge\(\)](#) (*mosromgr.mostypes.StoryAppend* method), 18
[merge\(\)](#) (*mosromgr.mostypes.StoryDelete* method), 20
[merge\(\)](#) (*mosromgr.mostypes.StoryInsert* method), 17
[merge\(\)](#) (*mosromgr.mostypes.StoryMove* method), 19
[merge\(\)](#) (*mosromgr.mostypes.StoryReplace* method), 15
[merge\(\)](#) (*mosromgr.mostypes.StorySend* method), 14
[message_id](#) (*mosromgr.moscollection.MosReader* property), 51
[message_id](#) (*mosromgr.mostypes.EAItemDelete* property), 32
[message_id](#) (*mosromgr.mostypes.EAItemInsert* property), 35
[message_id](#) (*mosromgr.mostypes.EAItemMove* property), 40
[message_id](#) (*mosromgr.mostypes.EAItemReplace* property), 30
[message_id](#) (*mosromgr.mostypes.EAItemSwap* property), 37
[message_id](#) (*mosromgr.mostypes.EAStoryDelete* property), 31
[message_id](#) (*mosromgr.mostypes.EAStoryInsert* property), 34
[message_id](#) (*mosromgr.mostypes.EAStoryMove* property), 38
[message_id](#) (*mosromgr.mostypes.EAStoryReplace* property), 29
[message_id](#) (*mosromgr.mostypes.EAStorySwap* property), 36
[message_id](#) (*mosromgr.mostypes.ElementAction* property), 47
[message_id](#) (*mosromgr.mostypes.ItemDelete* property), 23
[message_id](#) (*mosromgr.mostypes.ItemInsert* property), 24
[message_id](#) (*mosromgr.mostypes.ItemMoveMultiple* property), 25
[message_id](#) (*mosromgr.mostypes.ItemReplace* property), 26
[message_id](#) (*mosromgr.mostypes.MetadataReplace* property), 22
[message_id](#) (*mosromgr.mostypes.MosFile* property), 46
[message_id](#) (*mosromgr.mostypes.ReadyToAir* property), 28
[message_id](#) (*mosromgr.mostypes.RunningOrder* property), 13
[message_id](#) (*mosromgr.mostypes.RunningOrderControl* property), 43
[message_id](#) (*mosromgr.mostypes.RunningOrderEnd* property), 42
[message_id](#) (*mosromgr.mostypes.RunningOrderReplace* property), 41
[message_id](#) (*mosromgr.mostypes.StoryAppend* property), 18
[message_id](#) (*mosromgr.mostypes.StoryDelete* property), 20
[message_id](#) (*mosromgr.mostypes.StoryInsert* property), 17
[message_id](#) (*mosromgr.mostypes.StoryMove* property), 19
[message_id](#) (*mosromgr.mostypes.StoryReplace* property), 16
[message_id](#) (*mosromgr.mostypes.StorySend* property), 15
[MetadataReplace](#) (class in *mosromgr.mostypes*), 21
[module](#)
 [mosromgr.exc](#), 52
 [mosromgr.moscollection](#), 49
 [mosromgr.moselements](#), 47
 [mosromgr.mostypes](#), 11
 [mosromgr.utils](#), 51
[mos_id](#) (*mosromgr.moselements.Item* property), 48
[mos_object](#) (*mosromgr.moscollection.MosReader* property), 51
[mos_readers](#) (*mosromgr.moscollection.MosCollection* property), 50
[mos_type](#) (*mosromgr.moscollection.MosReader* property), 51
[MosCollection](#) (class in *mosromgr.moscollection*), 50
[MosCompletedMergeError](#), 53
[MosElement](#) (class in *mosromgr.moselements*), 49
[MosFile](#) (class in *mosromgr.mostypes*), 46
[MosInvalidXML](#), 53
[MosMergeError](#), 53
[MosMergeNonStrictWarning](#), 54
[MosReader](#) (class in *mosromgr.moscollection*), 51
[mosromgr.exc](#)
 module, 52
[mosromgr.moscollection](#)
 module, 49
[mosromgr.moselements](#)
 module, 47
[mosromgr.mostypes](#)
 module, 11
[mosromgr.utils](#)

module, 51
 mosromgr-detect command line option
 --help, 55
 -b --bucket-name bucket, 55
 -f --files [files ...], 55
 -h, 55
 -k --key key, 55
 -p --prefix prefix, 55
 -s --suffix suffix, 55
 mosromgr-inspect command line option
 --help, 56
 -b --bucket-name bucket, 56
 -f --files [files ...], 56
 -h, 56
 -k --key key, 56
 -p --prefix prefix, 56
 -s --suffix suffix, 56
 mosromgr-merge command line option
 --help, 57
 -b --bucket-name bucket, 57
 -f --files [files ...], 57
 -h, 57
 -i --incomplete, 57
 -n --non-strict, 57
 -o --outfile outfile, 57
 -p --prefix prefix, 57
 -s --suffix suffix, 57
 MosRoMgrException, 52
 MosRoMgrWarning, 53

N

note (*mosromgr.moselements.Item* property), 48

O

object_id (*mosromgr.moselements.Item* property), 48
 offset (*mosromgr.moselements.Story* property), 48

R

ReadyToAir (class in *mosromgr.mostypes*), 27
 remove_node() (in module *mosromgr.utils.xml*), 52
 replace_node() (in module *mosromgr.utils.xml*), 52
 ro (*mosromgr.moscollection.MosCollection* property), 50
 ro_id (*mosromgr.moscollection.MosCollection* property), 50
 ro_id (*mosromgr.moscollection.MosReader* property), 51
 ro_id (*mosromgr.mostypes.EAItemDelete* property), 33
 ro_id (*mosromgr.mostypes.EAItemInsert* property), 35
 ro_id (*mosromgr.mostypes.EAItemMove* property), 40
 ro_id (*mosromgr.mostypes.EAItemReplace* property), 30
 ro_id (*mosromgr.mostypes.EAItemSwap* property), 37
 ro_id (*mosromgr.mostypes.EAStoryDelete* property), 31
 ro_id (*mosromgr.mostypes.EAStoryInsert* property), 34
 ro_id (*mosromgr.mostypes.EAStoryMove* property), 38

ro_id (*mosromgr.mostypes.EAStoryReplace* property), 29
 ro_id (*mosromgr.mostypes.EAStorySwap* property), 36
 ro_id (*mosromgr.mostypes.ElementAction* property), 47
 ro_id (*mosromgr.mostypes.ItemDelete* property), 23
 ro_id (*mosromgr.mostypes.ItemInsert* property), 24
 ro_id (*mosromgr.mostypes.ItemMoveMultiple* property), 25
 ro_id (*mosromgr.mostypes.ItemReplace* property), 27
 ro_id (*mosromgr.mostypes.MetaDataReplace* property), 22
 ro_id (*mosromgr.mostypes.MosFile* property), 46
 ro_id (*mosromgr.mostypes.ReadyToAir* property), 28
 ro_id (*mosromgr.mostypes.RunningOrder* property), 13
 ro_id (*mosromgr.mostypes.RunningOrderControl* property), 44
 ro_id (*mosromgr.mostypes.RunningOrderEnd* property), 42
 ro_id (*mosromgr.mostypes.RunningOrderReplace* property), 41
 ro_id (*mosromgr.mostypes.StoryAppend* property), 18
 ro_id (*mosromgr.mostypes.StoryDelete* property), 20
 ro_id (*mosromgr.mostypes.StoryInsert* property), 17
 ro_id (*mosromgr.mostypes.StoryMove* property), 19
 ro_id (*mosromgr.mostypes.StoryReplace* property), 16
 ro_id (*mosromgr.mostypes.StorySend* property), 15
 ro_slug (*mosromgr.moscollection.MosCollection* property), 50
 ro_slug (*mosromgr.mostypes.MetaDataReplace* property), 22
 ro_slug (*mosromgr.mostypes.RunningOrder* property), 13
 ro_slug (*mosromgr.mostypes.RunningOrderReplace* property), 41
 RunningOrder (class in *mosromgr.mostypes*), 12
 RunningOrderControl (class in *mosromgr.mostypes*), 43
 RunningOrderEnd (class in *mosromgr.mostypes*), 42
 RunningOrderReplace (class in *mosromgr.mostypes*), 40

S

script (*mosromgr.moselements.Story* property), 48
 script (*mosromgr.mostypes.RunningOrder* property), 13
 script (*mosromgr.mostypes.RunningOrderReplace* property), 41
 slug (*mosromgr.moselements.Item* property), 48
 slug (*mosromgr.moselements.MosElement* property), 49
 slug (*mosromgr.moselements.Story* property), 48
 source_stories (*mosromgr.mostypes.StoryInsert* property), 17
 source_story (*mosromgr.mostypes.StoryMove* property), 19
 start_time (*mosromgr.moselements.Story* property), 48

- start_time (mosromgr.mostypes.RunningOrder property), 13
- start_time (mosromgr.mostypes.RunningOrderReplace property), 41
- stories (mosromgr.mostypes.EAStoryDelete property), 31
- stories (mosromgr.mostypes.EAStoryInsert property), 34
- stories (mosromgr.mostypes.EAStoryMove property), 38
- stories (mosromgr.mostypes.EAStoryReplace property), 29
- stories (mosromgr.mostypes.EAStorySwap property), 36
- stories (mosromgr.mostypes.RunningOrder property), 14
- stories (mosromgr.mostypes.RunningOrderReplace property), 41
- stories (mosromgr.mostypes.StoryAppend property), 18
- stories (mosromgr.mostypes.StoryDelete property), 20
- stories (mosromgr.mostypes.StoryReplace property), 16
- Story (class in mosromgr.moselements), 47
- story (mosromgr.mostypes.EAItemDelete property), 33
- story (mosromgr.mostypes.EAItemInsert property), 35
- story (mosromgr.mostypes.EAItemMove property), 40
- story (mosromgr.mostypes.EAItemReplace property), 30
- story (mosromgr.mostypes.EAItemSwap property), 37
- story (mosromgr.mostypes.EAStoryInsert property), 34
- story (mosromgr.mostypes.EAStoryMove property), 39
- story (mosromgr.mostypes.EAStoryReplace property), 29
- story (mosromgr.mostypes.ItemDelete property), 23
- story (mosromgr.mostypes.ItemInsert property), 24
- story (mosromgr.mostypes.ItemMoveMultiple property), 25
- story (mosromgr.mostypes.ItemReplace property), 27
- story (mosromgr.mostypes.RunningOrderControl property), 44
- story (mosromgr.mostypes.StoryReplace property), 16
- story (mosromgr.mostypes.StorySend property), 15
- StoryAppend (class in mosromgr.mostypes), 17
- StoryDelete (class in mosromgr.mostypes), 20
- StoryInsert (class in mosromgr.mostypes), 16
- StoryMove (class in mosromgr.mostypes), 18
- StoryNotFoundWarning, 54
- StoryReplace (class in mosromgr.mostypes), 15
- StorySend (class in mosromgr.mostypes), 14
- type (mosromgr.moselements.Item property), 48
- ## U
- UnknownMosFileType, 53
- ## X
- xml (mosromgr.moselements.Item property), 48
- xml (mosromgr.moselements.MosElement property), 49
- xml (mosromgr.moselements.Story property), 48
- xml (mosromgr.mostypes.EAItemDelete property), 33
- xml (mosromgr.mostypes.EAItemInsert property), 35
- xml (mosromgr.mostypes.EAItemMove property), 40
- xml (mosromgr.mostypes.EAItemReplace property), 30
- xml (mosromgr.mostypes.EAItemSwap property), 37
- xml (mosromgr.mostypes.EAStoryDelete property), 31
- xml (mosromgr.mostypes.EAStoryInsert property), 34
- xml (mosromgr.mostypes.EAStoryMove property), 39
- xml (mosromgr.mostypes.EAStoryReplace property), 29
- xml (mosromgr.mostypes.EAStorySwap property), 36
- xml (mosromgr.mostypes.ElementAction property), 47
- xml (mosromgr.mostypes.ItemDelete property), 23
- xml (mosromgr.mostypes.ItemInsert property), 24
- xml (mosromgr.mostypes.ItemMoveMultiple property), 25
- xml (mosromgr.mostypes.ItemReplace property), 27
- xml (mosromgr.mostypes.MetaDataReplace property), 22
- xml (mosromgr.mostypes.MosFile property), 46
- xml (mosromgr.mostypes.ReadyToAir property), 28
- xml (mosromgr.mostypes.RunningOrder property), 14
- xml (mosromgr.mostypes.RunningOrderControl property), 44
- xml (mosromgr.mostypes.RunningOrderEnd property), 43
- xml (mosromgr.mostypes.RunningOrderReplace property), 41
- xml (mosromgr.mostypes.StoryAppend property), 18
- xml (mosromgr.mostypes.StoryDelete property), 21
- xml (mosromgr.mostypes.StoryInsert property), 17
- xml (mosromgr.mostypes.StoryMove property), 19
- xml (mosromgr.mostypes.StoryReplace property), 16
- xml (mosromgr.mostypes.StorySend property), 15
- ## T
- target_story (mosromgr.mostypes.StoryInsert property), 17
- target_story (mosromgr.mostypes.StoryMove property), 19