

---

# **ACStore**

***Release 20211228***

**unknown**

**Aug 06, 2022**



# CONTENTS

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Installation instructions . . . . .	3
<b>2</b>	<b>acstore package</b>	<b>5</b>
2.1	Subpackages . . . . .	5
2.2	Submodules . . . . .	8
2.3	acstore.fake_store module . . . . .	8
2.4	acstore.interface module . . . . .	10
2.5	Module contents . . . . .	13
<b>3</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



ACStore, or Attribute Container Storage, provides a stand-alone implementation to read and write attribute container storage files.

The source code is available from the [project page](#).



## GETTING STARTED

To be able to use ACStore you first need to install it. There are multiple ways to install ACStore, check the following instructions for more detail.

### 1.1 Installation instructions

#### 1.1.1 pip

**Note that using pip outside virtualenv is not recommended since it ignores your systems package manager. If you aren't comfortable debugging package installation issues, this is not the option for you.**

Create and activate a virtualenv:

```
virtualenv acstoreenv  
cd acstoreenv  
source ./bin/activate
```

Upgrade pip and install ACStore dependencies:

```
pip install --upgrade pip  
pip install acstore
```

To deactivate the virtualenv run:

```
deactivate
```

#### 1.1.2 Ubuntu 18.04 and 20.04 LTS

To install ACStore from the [GIFT Personal Package Archive \(PPA\)](#):

```
sudo add-apt-repository ppa:gift/stable
```

Update and install ACStore:

```
sudo apt-get update  
sudo apt-get install python3-acstore
```

### 1.1.3 Windows

The [l2tbinaries](#) contains the necessary packages for running ACStore. l2tbinaries provides the following branches:

- main; branch intended for the “packaged release” of ACStore and dependencies;
- dev; branch intended for the “development release” of ACStore;
- testing; branch intended for testing newly created packages.

The l2tdevtools project provides [an update script](#) to ease the process of keeping the dependencies up to date.

The script requires [pywin32](#) and [Python WMI](#).

To install the release versions of the dependencies run:

```
set PYTHONPATH=.
C:\Python3\python.exe tools\update.py --preset acstore
```



## ACSTORE PACKAGE

### 2.1 Subpackages

#### 2.1.1 acstore.containers package

##### Submodules

##### acstore.containers.interface module

The attribute container interface.

**class** acstore.containers.interface.**AttributeContainer**

Bases: object

The attribute container interface.

This is the the base class for those object that exists primarily as a container of attributes with basic accessors and mutators.

The CONTAINER\_TYPE class attribute contains a string that identifies the container type, for example the container type “event” identifies an event object.

Attributes are public class members of an serializable type. Protected and private class members are not to be serialized, with the exception of those defined in `_SERIALIZABLE_PROTECTED_ATTRIBUTES`.

**CONTAINER\_TYPE = None**

**CopyFromDict**(*attributes*)

Copies the attribute container from a dictionary.

**Parameters**

**attributes** (*dict[str, object]*) – attribute values per name.

**CopyToDict**()

Copies the attribute container to a dictionary.

**Returns**

attribute values per name.

**Return type**

dict[str, object]

**GetAttributeNames**()

Retrieves the names of all attributes.

**Returns**

attribute names.

**Return type**

list[str]

**GetAttributeValuesHash()**

Retrieves a comparable string of the attribute values.

**Returns**

hash of comparable string of the attribute values.

**Return type**

int

**GetAttributeValuesString()**

Retrieves a comparable string of the attribute values.

**Returns**

comparable string of the attribute values.

**Return type**

str

**GetAttributes()**

Retrieves the attribute names and values.

Attributes that are set to None are ignored.

**Yields**

*tuple[str, object]* – attribute name and value.

**GetIdentifier()**

Retrieves the identifier.

The identifier is a storage specific value that should not be serialized.

**Returns**

an unique identifier for the container.

**Return type**

*AttributeContainerIdentifier*

**MatchesExpression(expression)**

Determines if an attribute container matches the expression.

**Parameters**

**expression** (*code/str*) – expression.

**Returns**

**True if the attribute container matches the expression, False**  
otherwise.

**Return type**

bool

**SetIdentifier(identifier)**

Sets the identifier.

The identifier is a storage specific value that should not be serialized.

**Parameters**

**identifier** (*AttributeContainerIdentifier*) – identifier.

**class** acstore.containers.interface.**AttributeContainerIdentifier**

Bases: object

The attribute container identifier.

The identifier is used to uniquely identify attribute containers. The value should be unique at runtime and in storage.

**CopyToString()**

Copies the identifier to a string representation.

**Returns**

unique identifier or None.

**Return type**

str

**acstore.containers.manager module**

This file contains the attribute container manager class.

**class** acstore.containers.manager.**AttributeContainersManager**

Bases: object

Class that implements the attribute container manager.

**classmethod** **CreateAttributeContainer**(*container\_type*)

Creates an instance of a specific attribute container type.

**Parameters**

**container\_type** (*str*) – container type.

**Returns**

an instance of attribute container.

**Return type**

*AttributeContainer*

**Raises**

**ValueError** – if the container type is not supported.

**classmethod** **DeregisterAttributeContainer**(*attribute\_container\_class*)

Deregisters an attribute container class.

The attribute container classes are identified based on their lower case container type.

**Parameters**

**attribute\_container\_class** (*type*) – attribute container class.

**Raises**

**KeyError** – if attribute container class is not set for the corresponding container type.

**classmethod** **GetContainerTypes**()

Retrieves the container types of the registered attribute containers.

**Returns**

container types.

**Return type**

list[str]

**classmethod** `GetSchema(container_type)`

Retrieves the schema of a registered attribute container.

**Parameters**

**container\_type** (*str*) – attribute container type.

**Returns**

**attribute container schema or an empty dictionary if**  
no schema available.

**Return type**

dict[str, str]

**Raises**

**ValueError** – if the container type is not supported.

**classmethod** `RegisterAttributeContainer(attribute_container_class)`

Registers a attribute container class.

The attribute container classes are identified based on their lower case container type.

**Parameters**

**attribute\_container\_class** (*type*) – attribute container class.

**Raises**

**KeyError** – if attribute container class is already set for the corresponding container type.

**classmethod** `RegisterAttributeContainers(attribute_container_classes)`

Registers attribute container classes.

The attribute container classes are identified based on their lower case container type.

**Parameters**

**attribute\_container\_classes** (*list[type]*) – attribute container classes.

**Raises**

**KeyError** – if attribute container class is already set for the corresponding container type.

## Module contents

## 2.2 Submodules

## 2.3 acstore.fake\_store module

Fake (in-memory only) attribute container store for testing.

**class** `acstore.fake_store.FakeAttributeContainerIdentifier(sequence_number)`

Bases: *AttributeContainerIdentifier*

Fake attribute container identifier.

**sequence\_number**

sequence number of the attribute container.

**Type**

int

**CopyToString()**

Copies the identifier to a string representation.

**Returns**

unique identifier or None.

**Return type**

str

**class acstore.fake\_store.FakeAttributeContainerStore**

Bases: *AttributeContainerStore*

Fake (in-memory only) attribute container store.

**Close()**

Closes the store.

**Raises**

- **IOError** – if the store is already closed.
- **OSError** – if the store is already closed.

**GetAttributeContainerByIdentifier**(*container\_type, identifier*)

Retrieves a specific type of container with a specific identifier.

**Parameters**

- **container\_type** (*str*) – container type.
- **identifier** (*AttributeContainerIdentifier*) – attribute container identifier.

**Returns**

attribute container or None if not available.

**Return type**

*AttributeContainer*

**GetAttributeContainerByIndex**(*container\_type, index*)

Retrieves a specific attribute container.

**Parameters**

- **container\_type** (*str*) – attribute container type.
- **index** (*int*) – attribute container index.

**Returns**

attribute container or None if not available.

**Return type**

*AttributeContainer*

**GetAttributeContainers**(*container\_type, filter\_expression=None*)

Retrieves a specific type of attribute containers.

**Parameters**

- **container\_type** (*str*) – attribute container type.
- **filter\_expression** (*Optional[str]*) – expression to filter the resulting attribute containers by.

**Yields**

*AttributeContainer* – attribute container.

**GetNumberOfAttributeContainers**(*container\_type*)

Retrieves the number of a specific type of attribute containers.

**Parameters**

**container\_type** (*str*) – attribute container type.

**Returns**

the number of containers of a specified type.

**Return type**

int

**HasAttributeContainers**(*container\_type*)

Determines if a store contains a specific type of attribute container.

**Parameters**

**container\_type** (*str*) – attribute container type.

**Returns**

**True if the store contains the specified type of attribute containers.**

**Return type**

bool

**Open**(\*\**kwargs*)

Opens the store.

**Raises**

- **IOError** – if the store is already opened.
- **OSError** – if the store is already opened.

## 2.4 acstore.interface module

The attribute container store interface.

**class** acstore.interface.**AttributeContainerStore**

Bases: object

Interface of an attribute container store.

**format\_version**

storage format version.

**Type**

int

**serialization\_format**

serialization format.

**Type**

str

**AddAttributeContainer**(*container*)

Adds a new attribute container.

**Parameters**

**container** (**AttributeContainer**) – attribute container.

**Raises**

- **OSError** – if the store cannot be written to.
- **IOError** – if the store cannot be written to.

**abstract Close()**

Closes the store.

**abstract GetAttributeContainerByIdentifier**(*container\_type, identifier*)

Retrieves a specific type of container with a specific identifier.

**Parameters**

- **container\_type** (*str*) – container type.
- **identifier** (*AttributeContainerIdentifier*) – attribute container identifier.

**Returns**

attribute container or None if not available.

**Return type**

*AttributeContainer*

**Raises**

- **IOError** – when the store is closed or if an unsupported identifier is provided.
- **OSError** – when the store is closed or if an unsupported identifier is provided.

**abstract GetAttributeContainerByIndex**(*container\_type, index*)

Retrieves a specific attribute container.

**Parameters**

- **container\_type** (*str*) – attribute container type.
- **index** (*int*) – attribute container index.

**Returns**

attribute container or None if not available.

**Return type**

*AttributeContainer*

**Raises**

- **IOError** – when the store is closed.
- **OSError** – when the store is closed.

**abstract GetAttributeContainers**(*container\_type, filter\_expression=None*)

Retrieves a specific type of attribute containers.

**Parameters**

- **container\_type** (*str*) – attribute container type.
- **filter\_expression** (*Optional[str]*) – expression to filter the resulting attribute containers by.

**Returns**

attribute container generator.

**Return type**

generator(*AttributeContainer*)

**Raises**

- **IOError** – when the store is closed.
- **OSError** – when the store is closed.

**abstract GetNumberOfAttributeContainers**(*container\_type*)

Retrieves the number of a specific type of attribute containers.

**Parameters****container\_type** (*str*) – attribute container type.**Returns**

the number of containers of a specified type.

**Return type**

int

**abstract HasAttributeContainers**(*container\_type*)

Determines if a store contains a specific type of attribute container.

**Parameters****container\_type** (*str*) – attribute container type.**Returns****True if the store contains the specified type of attribute containers.****Return type**

bool

**abstract Open**(\*\**kwargs*)

Opens the store.

**SetStorageProfiler**(*storage\_profiler*)

Sets the storage profiler.

**Parameters****storage\_profiler** (*StorageProfiler*) – storage profiler.**UpdateAttributeContainer**(*container*)

Updates an existing attribute container.

**Parameters****container** (*AttributeContainer*) – attribute container.**Raises**

- **OSError** – if the store cannot be written to.
- **IOError** – if the store cannot be written to.



## 2.5 Module contents

Attribute Container Storage (ACStore).

ACStore, or Attribute Container Storage, provides a stand-alone implementation to read and write plaso storage files.



## INDICES AND TABLES

- genindex
- modindex



## PYTHON MODULE INDEX

### a

`acstore`, 13

`acstore.containers`, 8

`acstore.containers.interface`, 5

`acstore.containers.manager`, 7

`acstore.fake_store`, 8

`acstore.interface`, 10



## INDEX

### A

acstore  
  module, 13  
acstore.containers  
  module, 8  
acstore.containers.interface  
  module, 5  
acstore.containers.manager  
  module, 7  
acstore.fake\_store  
  module, 8  
acstore.interface  
  module, 10  
AddAttributeContainer() (ac-  
  store.interface.AttributeContainerStore  
  method), 10  
AttributeContainer (class in ac-  
  store.containers.interface), 5  
AttributeContainerIdentifier (class in ac-  
  store.containers.interface), 7  
AttributeContainersManager (class in ac-  
  store.containers.manager), 7  
AttributeContainerStore (class in acstore.interface),  
  10

### C

Close() (acstore.fake\_store.FakeAttributeContainerStore  
  method), 9  
Close() (acstore.interface.AttributeContainerStore  
  method), 11  
CONTAINER\_TYPE (acstore.containers.interface.AttributeContainer  
  attribute), 5  
CopyFromDict() (acstore.containers.interface.AttributeContainer  
  method), 5  
CopyToDict() (acstore.containers.interface.AttributeContainer  
  method), 5  
CopyToString() (acstore.containers.interface.AttributeContainerIdentifier  
  method), 7  
CopyToString() (acstore.fake\_store.FakeAttributeContainerIdentifier  
  method), 8  
CreateAttributeContainer() (ac-  
  store.containers.manager.AttributeContainersManager

class method), 7

### D

DeregisterAttributeContainer() (ac-  
  store.containers.manager.AttributeContainersManager  
  class method), 7

### F

FakeAttributeContainerIdentifier (class in ac-  
  store.fake\_store), 8  
FakeAttributeContainerStore (class in ac-  
  store.fake\_store), 9  
format\_version (acstore.interface.AttributeContainerStore  
  attribute), 10

### G

GetAttributeContainerByIdentifier() (ac-  
  store.fake\_store.FakeAttributeContainerStore  
  method), 9  
GetAttributeContainerByIdentifier() (ac-  
  store.interface.AttributeContainerStore  
  method), 11  
GetAttributeContainerByIndex() (ac-  
  store.fake\_store.FakeAttributeContainerStore  
  method), 9  
GetAttributeContainerByIndex() (ac-  
  store.interface.AttributeContainerStore  
  method), 11  
GetAttributeContainers() (ac-  
  store.fake\_store.FakeAttributeContainerStore  
  method), 9  
GetAttributeContainers() (ac-  
  store.interface.AttributeContainerStore  
  method), 11  
GetAttributeNames() (ac-  
  store.containers.interface.AttributeContainer  
  method), 5  
GetAttributes() (ac-  
  store.containers.interface.AttributeContainer  
  method), 6  
GetAttributeValuesHash() (ac-  
  store.containers.interface.AttributeContainer

- method*), 6
- GetAttributeValuesString() (*ac-store.containers.interface.AttributeContainer method*), 6
- GetContainerTypes() (*ac-store.containers.manager.AttributeContainersManager class method*), 7
- GetIdentifier() (*ac-store.containers.interface.AttributeContainer method*), 6
- GetNumberOfAttributeContainers() (*ac-store.fake\_store.FakeAttributeContainerStore method*), 9
- GetNumberOfAttributeContainers() (*ac-store.interface.AttributeContainerStore method*), 12
- GetSchema() (*acstore.containers.manager.AttributeContainersManager class method*), 7
- ## H
- HasAttributeContainers() (*ac-store.fake\_store.FakeAttributeContainerStore method*), 10
- HasAttributeContainers() (*ac-store.interface.AttributeContainerStore method*), 12
- ## M
- MatchesExpression() (*ac-store.containers.interface.AttributeContainer method*), 6
- module
- acstore, 13
  - acstore.containers, 8
  - acstore.containers.interface, 5
  - acstore.containers.manager, 7
  - acstore.fake\_store, 8
  - acstore.interface, 10
- ## O
- Open() (*acstore.fake\_store.FakeAttributeContainerStore method*), 10
- Open() (*acstore.interface.AttributeContainerStore method*), 12
- ## R
- RegisterAttributeContainer() (*ac-store.containers.manager.AttributeContainersManager class method*), 8
- RegisterAttributeContainers() (*ac-store.containers.manager.AttributeContainersManager class method*), 8
- ## S
- sequence\_number (*ac-store.fake\_store.FakeAttributeContainerIdentifier attribute*), 8
- serialization\_format (*ac-store.interface.AttributeContainerStore attribute*), 10
- SetIdentifier() (*ac-store.containers.interface.AttributeContainer method*), 6
- SetStorageProfiler() (*ac-store.interface.AttributeContainerStore method*), 12
- ## U
- UpdateAttributeContainer() (*ac-store.interface.AttributeContainerStore method*), 12