

---

# **Pacifica Archive Interface Documentation**

**David Brown**

**May 13, 2022**



---

## Contents:

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Installation in Virtual Environment . . . . .	3
<b>2</b>	<b>Configuration</b>	<b>5</b>
2.1	CherryPy Configuration File . . . . .	5
2.2	Service Configuration File . . . . .	5
2.3	ID Mapping to File Names . . . . .	6
2.4	Running It . . . . .	6
<b>3</b>	<b>Post Deployment Testing</b>	<b>7</b>
3.1	Variables . . . . .	7
3.2	Running . . . . .	7
<b>4</b>	<b>Example Usage</b>	<b>9</b>
4.1	Verify working . . . . .	9
4.2	Put a File . . . . .	9
4.3	Get a File . . . . .	10
4.4	Status a File . . . . .	10
4.5	Stage a File . . . . .	10
4.6	Move a File . . . . .	10
<b>5</b>	<b>Extending Supported Backends</b>	<b>13</b>
5.1	Create a backend directory . . . . .	13
5.2	Create Classes that Implement the Abstract Backend Class methods . . . . .	13
5.3	Update Backend Factory . . . . .	13
5.4	Update Interface Server . . . . .	13
<b>6</b>	<b>Archive Interface Python Module</b>	<b>15</b>
6.1	Backends Python Module . . . . .	15
6.2	Configuration Python Module . . . . .	24
6.3	Exception Python Module . . . . .	24
6.4	Globals Python Module . . . . .	24
6.5	ID to Filename Python Module . . . . .	24
6.6	Rest Python Module . . . . .	25
6.7	Utils Python Module . . . . .	26
6.8	WSGI Python Module . . . . .	26

<b>7 Indices and tables</b>	<b>27</b>
<b>Python Module Index</b>	<b>29</b>
<b>Index</b>	<b>31</b>

The Pacifica Archive Interface service provides a REST API to a hierarchical storage management (HSM) system. The service provides Pacifica the ability to store files for the long term in a scalable cheap storage solution.



The Pacifica software is available through PyPi so creating a virtual environment to install is what is shown below. Please keep in mind compatibility with the Pacifica Core services.

### 1.1 Installation in Virtual Environment

These installation instructions are intended to work on both Windows, Linux, and Mac platforms. Please keep that in mind when following the instructions.

Please install the appropriate tested version of Python for maximum chance of success.

#### 1.1.1 Linux and Mac Installation

```
mkdir ~/.virtualenvs
python -m virtualenv ~/.virtualenvs/pacifica
. ~/.virtualenvs/pacifica/bin/activate
pip install pacifica-archiveinterface
```

#### 1.1.2 Windows Installation

This is done using PowerShell. Please do not use Batch Command.

```
mkdir "$Env:LOCALAPPDATA\virtualenvs"
python.exe -m virtualenv "$Env:LOCALAPPDATA\virtualenvs\pacifica"
& "$Env:LOCALAPPDATA\virtualenvs\pacifica\Scripts\activate.ps1"
pip install pacifica-archiveinterface
```





The configuration of the Archive Interface service is done in two files. The REST API utilizes [CherryPy](#) and review of their [configuration documentation](#) is recommended. The service configuration file is a INI formatted file containing configuration for accessing the HSM backend.

### 2.1 CherryPy Configuration File

An example of Archive Interface server CherryPy configuration:

```
[global]
log.screen: True
log.access_file: 'access.log'
log.error_file: 'error.log'
server.socket_host: '0.0.0.0'
server.socket_port: 8080

[/]
request.dispatch: cherrypy.dispatch.MethodDispatcher()
tools.response_headers.on: True
tools.response_headers.headers: [('Content-Type', 'application/json')]
```

### 2.2 Service Configuration File

The service configuration is an INI file and an example is as follows:

```
[posix]
; use id2filename method when accessing a posix endpoint
use_id2filename = false

[hpss]
```

(continues on next page)

(continued from previous page)

```
; IBM HPSS HSM settings
auth = /var/hpss/etc/hpss.unix.keytab
sitename = example.com
user = hpss.unix

[hsm_sideband]
; Oracle HSM Sideband database settings (MySQL)
sam_qfs_prefix = /tmp/path
schema = schema_name
host = host
user = user
password = pass
port = 3306
```

## 2.3 ID Mapping to File Names

The Pacifica software depends on a flat ID space for indexing files. This needs to map to filenames on the backend storage in a nice way. To limit the number of files in a single directory (or number of directories in a directory) we use the algorithm in `archiveinterface.id2filename`. This takes a number and breaks it into bytes. Each byte is then represented in hex and used to build the directory tree.

For example `id2filename(12345)` becomes `/39/3039` on the backend file system. See more documentation in the *id2filename module*.

## 2.4 Running It

There are two ways of running the archive interface, using CherryPy builtin server and uWSGI to run the server.

CherryPy does provide a server and this works for some workloads.

```
pacifica-archiveinterface -t posix -p 8080 -a 127.0.0.1 --prefix /path
```

uWSGI is recommended for this service as it performs a lot better for higher throughput workloads.

```
export PAI_BACKEND_TYPE=posix
export PAI_PREFIX=/path
uwsgi --http-socket :8080 --master -p 8 --module pacifica.archiveinterface.wsgi
```

---

## Post Deployment Testing

---

Inside the `post_deployment_test` directory there is a file called `deployment_test.py`. This file will run a series of tests against a deployed archive interface. The tests are ordered so that they post, stage, status, and get files properly. There are a few global variables at the top of the file that need to be adjusted to each deployment.

### 3.1 Variables

```
export ARCHIVE_URL='http://127.0.0.1:8080/'
export LARGE_FILE_SIZE=$(( 1024 * 1024 * 1024 ))
export MANY_FILES_TEST_COUNT=1000
```

- `ARCHIVE_URL` is the URL to the newly deployed archive interface
- `LARGE_FILE_SIZE` is the size of the large file to test with (default 1Gib)
- `MANY_FILES_TEST_COUNT` is the number of small files to spam (default 1000)

### 3.2 Running

```
pytest -v post_deployment_tests/deployment_test.py
```

Output will be the status of the tests against the archive interface



---

## Example Usage

---

### 4.1 Verify working

To verify the system is working do a GET against the system with no id specified.

```
curl -X GET http://127.0.0.1:8080
```

Sample output:

```
{
  "message": "Pacifica Archive Interface Up and Running"
}
```

### 4.2 Put a File

The path in the URL should be only an integer specifying a unique file in the archive. Sending a different file to the same URL will over-write the contents of the previous file. Setting the Last- Modified header sets the mtime of the file in the archive and is required.

```
curl -X PUT -H 'Last-Modified: Sun, 06 Nov 1994 08:49:37 GMT' --upload-file /tmp/foo.
↳txt http://127.0.0.1:8080/12345
```

Sample output:

```
{
  "message": "File added to archive",
  "total_bytes": "24"
}
```

## 4.3 Get a File

The HTTP GET method is used to get the contents of the specified file.

```
curl -o /tmp/foo.txt http://127.0.0.1:8080/12345
```

Sample output (without -o option): “Document Contents”

## 4.4 Status a File

The HTTP HEAD method is used to get a JSON document describing the status of the file. The status includes, but is not limited to, the size, mtime, ctime, whether its on disk or tape. The values can be found within the headers.

```
curl -I -X HEAD http://127.0.0.1:8080/12345
```

Sample output:

```
HTTP/1.0 204 No Content
Date: Fri, 07 Oct 2016 19:51:37 GMT
Server: WSGIServer/0.1 Python/2.7.5
X-Pacifica-Message: File was found
X-Pacifica-File: /tmp/12345
Content-Length: 18
Last-Modified: 1473806059.29
X-Pacifica-Ctime: 1473806059.29
X-Pacifica-Bytes-Per-Level: (18L,)
X-Pacifica-File-Storage-Media: disk
Content-Type: application/json
```

## 4.5 Stage a File

The HTTP POST method is used to stage a file for use. In posix this equates to a no-op on hpss it stages the file to the disk drive.

```
curl -X POST -d '' http://127.0.0.1:8080/12345
```

Sample Output:

```
{
  "file": "/12345",
  "message": "File was staged"
}
```

## 4.6 Move a File

The HTTP PATCH method is used to move a file. The upload file contains the path to current file on archive The Id at the end of the url is where the file will be moved to

```
curl -X PATCH -H 'Content-Type: application/json' http://127.0.0.1:8080/123456 -d'{
  "path": "/tmp/12345"
}'
```

Sample Output:

```
{
  "message": "File Moved Successfully"
}
```





---

## Extending Supported Backends

---

### 5.1 Create a backend directory

Under `pacifica/archiveinterface/backends` add a directory for the new backend type

### 5.2 Create Classes that Implement the Abstract Backend Class methods

Abstract backend classes can be found under: `pacifica/archiveinterface/backends/abstract`. Descriptions of all the methods that need to be abstracted exist in the comments above the class.

### 5.3 Update Backend Factory

Update the archive backend factory found here: `pacifica/archiveinterface/backends/factory.py`. In this file is a `load_backend_archive()` method. This method needs to have its logic extended to support the new backend type. This also entails loading the appropriate files for this backend using `import`.

### 5.4 Update Interface Server

Update the `main()` method to support the new backend choice. File located: `pacifica/archiveinterface/__main__.py`. In this file the type argument is defined with its supported types. Need to extend that array to include the new backend type.



---

## Archive Interface Python Module

---

### 6.1 Backends Python Module

#### 6.1.1 Abstract Python Module

Module for the Abstract Classes.

Abstract Backend Module.

Module that has the Abstract class for Archive Backends Any new backends need to inherit from this class and implement its methods. If the methods are not implemented in the child, the child object will not be able to be instantiated.

**class** `pacificarchiveinterface.backends.abstract.archive.AbstractBackendArchive` (*prefix*)  
Abstract Base Class for Archive Backends.

**\_\_init\_\_** (*prefix*)  
Constructor to build backend archive.

**close** ()  
Close File.

Method that closes an open file for the backend archive that implements this class.

**open** (*filepath, mode*)  
Open File.

Method that opens a file for the backend archive that implements this class Should return a file like object, most likely self. This method is also responsible for making sure the dirname of the filepath exists before trying to open.

**patch** (*file\_id, old\_path*)  
Move a file.

**read** (*blocksize*)  
Read File.

Method that reads an open file for the backend archive that implements this class and returns the contents.

**remove** ()

Remove a file.

**seek** (*offset*)

Seek to an offset in the file.

Method that seeks to an offset in the file for the backend archive that implements this class.

**set\_file\_permissions** ()

Set permissions for File.

Method that sets a files permissions for the backend archive that implements this class.

**set\_mod\_time** (*mod\_time*)

Set Modification Time for File.

Method that sets a files mod time for the backend archive that implements this class.

**stage** ()

Stage File.

Method that stages a file for the the backend that implements this class Stage moves a file to an appropriate location to be downloaded.

**status** ()

Return status of the file.

Method that gets the status of a file in the archive Needs to return an implemented object of the abstract\_status\_class The abstract\_status\_class should be implemented for each backend type.

**write** (*buf*)

Write File.

Method that writes an open file for the backend archive that implements this class.

Module that has the abstract class for a file's status.

```
class pacifica.archiveinterface.backends.abstract.status.AbstractStatus (mtime,  
ctime,  
bytes_per_level,  
file-  
size)
```

Abstract Base Class for Status.

Child backend objects need to implement the following methods to allow file status to function.

**\_\_init\_\_** (*mtime, ctime, bytes\_per\_level, filesize*)

Constructor for AbstractClass.

Implemented versions of this class need to set all the attributes defined in this abstract class.

**bytes\_per\_level** = None

**ctime** = None

**define\_levels** ()

Defined list of levels.

Method that defines the storage levels in the archive backend. So a backend archive with a disk, tape, and error drive will return the following ["disk", "tape", "error"].

**defined\_levels** = None

**file\_storage\_media** = None

**filepath** = None

**filesize** = None

**find\_file\_storage\_media** ()

Find File Storage Media.

Method that finds the media the file in the archive backend is stored on. Usually disk or tape.

**mtime** = None

**set\_filepath** (*filepath*)

Set File Path.

Method that sets the filepath class attribute. Used to return the correct status of a file.

## 6.1.2 HPSS Python Module

HPSS Backend Module.

Module that implements the Abstract backend archive for an hpss backend.

**class** `pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive` (*prefix*)

The HPSS implementation of the backend archive.

**\_\_init\_\_** (*prefix*)

Constructor for the HPSS Backend Archive.

**static** **\_check\_rcode** (*rcode, msg*)

Check if rcode is < 0 and raise error.

**authenticate** ()

Authenticate the user with the hpss system.

**close** ()

Close an HPSS File.

**open** (*filepath, mode*)

Open an hpss file.

**patch** (*file\_id, old\_path*)

Move a hpss file.

**read** (*blocksize*)

Read a file from the hpss archive.

**remove** ()

Remove the file for an HPSS file.

**seek** (*offset*)

Seek in the file to offset.

**set\_file\_permissions** ()

Set the file permissions for an hpss archive file.

**set\_mod\_time** (*mod\_time*)

Set the mod time for an hpss archive file.

**stage** ()

Stage an hpss file to the top level drive.

**status** ()

Get the status of a file in the hpss archive.

**write** (*buf*)

Write a file to the hpss archive.

`pacifica.archiveinterface.backends.hpss.archive.path_info_munge` (*filepath*)

Munge the path for this filetype.

HPSS Extended File Module.

Module that holds the class to the interface for the hpss c extensions.

**class** `pacifica.archiveinterface.backends.hpss.extended.HpssExtended` (*filepath*)  
Provide the interface for the hpss ctypes.

`__init__` (*filepath*)

Constructor for the HPSS Extended File type.

**makedirs** ()

Recursively make the directories for the filepath.

**parse\_latency** (*latency\_tuple*)

Parse the latency tuple.

Parse the tuple returned by the c extension into the correct latency.

**ping\_core** (*sitename*)

Ping the Core server to see if its still active.

**set\_mod\_time** (*mod\_time*)

Use extensions to set the mod time on an hpss file.

**stage** ()

Stage an hpss file.

Do this to move the file to disk doesnt need to return anything. will throw exception on error however.

**status** ()

Get the status of a file.

If it is on tape or disk Found the documentation for this in the hpss programmers reference section 2.3.6.2.8  
“Get Extended Attributes”.

HPSS Status Module.

Module that implements the Abstract Status class for the hpss archive backend type.

**class** `pacifica.archiveinterface.backends.hpss.status.HpssStatus` (*mtime*, *ctime*,  
*bytes\_per\_level*,  
*filesize*)

HPSS Status Class.

Class for handling hpss status pieces needs mtime,ctime, bytes per level array.

`__init__` (*mtime*, *ctime*, *bytes\_per\_level*, *filesize*)

HPSS Status Constructor.

`_disk` = 'disk'

`_error` = 'error'

`_tape` = 'tape'

**define\_levels** ()

Set up what each level definition means.

**find\_file\_storage\_media** ()

Set if file is on disk or tape.

**set\_filepath** (*filepath*)  
Set the filepath that the status is for.

### 6.1.3 Oracle HSM Sideband Python Module

Oracle HSM Side Band Database Module.

HSM Sideband Backend Archive Module.

Module that implements the abstract\_backend\_archive class for a HSM Sideband backend.

**class** pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArch  
HSM Sideband Backend Archive Class.

Class that implements the abstract base class for the hsm sideband archive interface backend.

**\_\_init\_\_** (*prefix*)  
Constructor for HSM Sideband Backend Archive.

**close** ()  
Close a HSM Sideband file.

**open** (*filepath, mode*)  
Open a hsm sideband file.

**patch** (*file\_id, old\_path*)  
Move a hsm file.

**read** (*blocksize*)  
Read a HSM Sideband file.

**remove** ()  
Remove the file for a posix file.

**seek** (*offset*)  
Seek in the file to the offset.

**set\_file\_permissions** ()  
Set the file permissions for a posix file.

**set\_mod\_time** (*mod\_time*)  
Set the mod time on a HSM file.

**stage** ()  
Stage a HSM Sideband file.

**status** ()  
Get the status of a HSM Sideband file.

**write** (*buf*)  
Write a HSM Sideband file to the archive.

pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.path\_info\_munge (*filepath*)  
Munge the path for this filetype.

Module that allows for the extension of the hsm sideband archive.

pacifica.archiveinterface.backends.oracle\_hsm\_sideband.extended\_file\_factory.extended\_hsms

Return appropriate binary io object with additional methods.

ORM for the sideband database.

```
class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.BaseModel (*args,  
                                                                    **kwargs)
```

Base class models inherit from.

Has Connection pieces.

**DoesNotExist**

alias of BaseModelDoesNotExist

**\_meta** = <peewee.Metadata object>

**\_schema** = <peewee.SchemaManager object>

**classmethod database\_close** ()

Close the database connection.

**classmethod database\_connect** ()

Make sure database is connected.

Dont reopen connection.

**id** = <AutoField: BaseModel.id>

**reload** ()

Reload my current state from the DB.

```
class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.SamArchive (*args,  
                                                                    **kwargs)
```

Model for sam\_archive table in the sideband database.

**DoesNotExist**

alias of SamArchiveDoesNotExist

**\_meta** = <peewee.Metadata object>

**\_schema** = <peewee.SchemaManager object>

**copy** = <IntegerField: SamArchive.copy>

**create\_time** = <IntegerField: SamArchive.create\_time>

**gen** = <IntegerField: SamArchive.gen>

**ino** = <IntegerField: SamArchive.ino>

**media\_type** = <CharField: SamArchive.media\_type>

**offset** = <IntegerField: SamArchive.offset>

**position** = <BigIntegerField: SamArchive.position>

**seq** = <IntegerField: SamArchive.seq>

**size** = <BigIntegerField: SamArchive.size>

**stale** = <IntegerField: SamArchive.stale>

**vsn** = <CharField: SamArchive.vsn>

```
class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.SamFile (*args,  
                                                                    **kwargs)
```

Model for sam\_file table in the sideband database.

**DoesNotExist**

alias of SamFileDoesNotExist

**\_meta** = <peewee.Metadata object>



```

_schema = <peewee.SchemaManager object>
gen = <IntegerField: SamFile.gen>
ino = <IntegerField: SamFile.ino>
name = <CharField: SamFile.name>
name_hash = <IntegerField: SamFile.name_hash>
p_gen = <IntegerField: SamFile.p_gen>
p_ino = <IntegerField: SamFile.p_ino>

```

```

class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.SamInode(*args,
                                                                    **kwargs)

```

Model for sam\_inode table in the sideband database.

**DoesNotExist**

alias of SamInodeDoesNotExist

```

_meta = <peewee.Metadata object>
_schema = <peewee.SchemaManager object>
create_time = <IntegerField: SamInode.create_time>
csum = <CharField: SamInode.csum>
gen = <IntegerField: SamInode.gen>
gid = <IntegerField: SamInode.gid>
ino = <IntegerField: SamInode.ino>
modify_time = <IntegerField: SamInode.modify_time>
online = <IntegerField: SamInode.online>
size = <BigIntegerField: SamInode.size>
type = <IntegerField: SamInode.type>
uid = <IntegerField: SamInode.uid>

```

```

class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.SamPath(*args,
                                                                    **kwargs)

```

Model for sam\_path table in the sideband database.

**DoesNotExist**

alias of SamPathDoesNotExist

```

_meta = <peewee.Metadata object>
_schema = <peewee.SchemaManager object>
gen = <IntegerField: SamPath.gen>
ino = <IntegerField: SamPath.ino>
path = <CharField: SamPath.path>

```

```

class pacifica.archiveinterface.backends.oracle_hsm_sideband.orm.SamVersion(*args,
                                                                    **kwargs)

```

Model for sam\_version table in the sideband database.

**DoesNotExist**

alias of SamVersionDoesNotExist

```
_meta = <peewee.Metadata object>
_schema = <peewee.SchemaManager object>
id = <IntegerField: SamVersion.id>
version = <FloatField: SamVersion.version>
```

Module that implements the Abstract Status class.

For the oracle hsm sideband archive backend type.

```
class pacifica.archiveinterface.backends.oracle_hsm_sideband.status.HsmSidebandStatus (mtime,
                                                                                       ctime,
                                                                                       bytes_per_level,
                                                                                       file_size)
```

Class for handling hsmSideband status pieces.

Needs mtime,ctime, bytes per level array

```
__init__ (mtime, ctime, bytes_per_level, filesize)
    Constructor to build the object.
```

```
_disk = 'disk'
```

```
_tape = 'tape'
```

```
define_levels ()
    Set up what each level definition means.
```

```
find_file_storage_media ()
    Get the file storage media.
```

Should always be disk for hsmSideband.

```
set_filepath (filepath)
    Set the filepath that the status is for.
```

## 6.1.4 Posix Python Module

Posix Module with ExtendedFile object.

Posix Backend Archive Module.

Module that implements the abstract\_backend\_archive class for a posix backend.

```
class pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive (prefix)
    Posix Backend Archive Class.
```

Class that implements the abstract base class for the posix archive interface backend.

```
__init__ (prefix)
    Constructor for Posix Backend Archive.
```

```
close ()
    Close a posix file.
```

```
open (filepath, mode)
    Open a posix file.
```

```
patch (file_id, old_path)
    Move a posix file.
```

**read** (*blocksize*)  
Read a posix file.

**remove** ()  
Remove the file permissions for a posix file.

**seek** (*offset*)  
Seek in the file to the offset.

**set\_file\_permissions** ()  
Set the file permissions for a posix file.

**set\_mod\_time** (*mod\_time*)  
Set the mod time on a posix file.

**stage** ()  
Stage a posix file (no-opt essentially).

**status** ()  
Get the status of a posix file.

**write** (*buf*)  
Write a posix file to the archive.

Extended File Object Module.

Module that Extends the functionality of the base file object

to import:

```
>>> import ExtendedFile
>>> from extendedfile import ExtendedFile
>>> ExtendedFile(path, mode)
```

`pacifica.archiveinterface.backends.posix.extendedfile.extended_file_factory` (*filepath*,  
*mode*)

Return appropriate binary io object with additional methods.

Posix Status Module.

Module that implements the Abstract Status class for the posix archive backend type.

```
class pacifica.archiveinterface.backends.posix.status.PosixStatus (mtime,  
                                                             ctime,  
                                                             bytes_per_level,  
                                                             filesize)
```

Posix Status Class.

Class for handling posix status pieces needs *mtime*, *ctime*, bytes per level array.

```
__init__ (mtime, ctime, bytes_per_level, filesize)  
    Constructor for posix status class.
```

```
_disk = 'disk'
```

```
define_levels ()  
    Set up what each level definition means.
```

```
find_file_storage_media ()  
    Get the file storage media. Showed always be disk for posix.
```

```
set_filepath (filepath)  
    Set the filepath that the status is for.
```

### 6.1.5 Factory Python Module

Factory for returning a Archive backend.

New Backends must be added to the `__share_classes` list and that class needs to be imported in

Call the factory like the following: `FACTORY = ArchiveBackendFactory() BACKEND = FACTORY.get_backend_archive(type, prefix)`

**class** `pacifica.archiveinterface.backends.factory.ArchiveBackendFactory`  
Factory Class for Archive Backends.

**get\_backend\_archive** (*name, prefix*)  
Method for creating an instance of the backend archive.

**load\_backend\_archive** (*name*)  
Method for loading in the correct backend type.

Only want to load backend type being used.

**share\_classes** = {}

Backends module contains archive interface backends specifics.

## 6.2 Configuration Python Module

Configuration reading and validation module.

`pacifica.archiveinterface.config.get_config()`  
Return the ConfigParser object with defaults set.

## 6.3 Exception Python Module

Module with the Archive Interface Error class.

**exception** `pacifica.archiveinterface.exception.ArchiveInterfaceError`  
ArchiveInterfaceError.

Basic exception class for this module. Will be used to throw exceptions up to the top level of the application

## 6.4 Globals Python Module

Global configuration options expressed in environment variables.

## 6.5 ID to Filename Python Module

Module to convert an integer id to a filepath for storage system.

Mathematically, this is a *bijjective* set of methods. This means that running `filename2id(id2filename(ID))` must return the same *ID* passed and conversely `id2filename(filename2id(PATH))` must return the same *PATH* passed.

Example usage:

```
>>> id2filename(1234)
'/d2/4d2'
>>> filename2id('/d2/4d2')
1234
```

`pacifica.archiveinterface.id2filename.filename2id(path)`  
Convert a filepath to and ID.

`pacifica.archiveinterface.id2filename.id2filename(fileid)`  
Algorithm for getting filepath from an integer id.

## 6.6 Rest Python Module

Class for the archive interface.

Allows API to file interactions for passed in archive backends.

**class** `pacifica.archiveinterface.rest_generator.ArchiveInterfaceGenerator` (*archive*)  
Archive Interface Generator.

Defines the methods that can be used on files for request types.

**DELETE** (*filepath*)

Delete a file from WSGI request.

Delete the file specified in the request to disk.

**GET** (*\*args, \*\*kwargs*)

Get a file from WSGI request.

Gets a file specified in the request and writes back the data.

**HEAD** (*filepath*)

Get the file status from WSGI request.

Gets the status of a file specified in the request.

**PATCH** (*filepath*)

Move a file from the original path to the new one specified.

**POST** (*filepath*)

Stage a file from WSGI request.

Stage the file specified in the request to disk.

**PUT** (*filepath*)

Write a file from WSGI requests.

Writes a file passed in the request to the archive.

**\_\_init\_\_** (*archive*)

Create an archive interface generator.

**static** `_byte_range_get` (*archivefile, byte\_range*)

Perform a byte range get and return generator.

**exposed = True**

`pacifica.archiveinterface.rest_generator.error_page_default` (*\*\*kwargs*)  
The default error page should always enforce json.

## 6.7 Utils Python Module

Group of utility functions.

Used in various parts of the archive interface.

`pacifica.archiveinterface.archive_utils.bytes_type(unicode_obj)`  
Convert the unicode object into bytes.

`pacifica.archiveinterface.archive_utils.file_status(status, response)`  
Response for when file is on the hpss system.

`pacifica.archiveinterface.archive_utils.get_http_modified_time(env)`  
Get the modified time from the request in unix timestamp.  
Returns current time if no time was passed.

`pacifica.archiveinterface.archive_utils.un_abs_path(path_name)`  
Remove absolute path piece.

## 6.8 WSGI Python Module

Pacifica Archive Interface.

This is the main program that starts the WSGI server.

The core of the server code is in `archive_interface.py`.

Any new Backends added need to have the `type` argument extended to support the new Backend Archie type Pacifica ArchiveInterface module.

# CHAPTER 7

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





### p

`pacifica.archiveinterface`, 26

`pacifica.archiveinterface.archive_utils`, 26

`pacifica.archiveinterface.backends`, 24

`pacifica.archiveinterface.backends.abstract`, 15

`pacifica.archiveinterface.backends.abstract.archive`, 15

`pacifica.archiveinterface.backends.abstract.status`, 16

`pacifica.archiveinterface.backends.factory`, 24

`pacifica.archiveinterface.backends.hpss`, 17

`pacifica.archiveinterface.backends.hpss.archive`, 17

`pacifica.archiveinterface.backends.hpss.extended`, 18

`pacifica.archiveinterface.backends.hpss.status`, 18

`pacifica.archiveinterface.backends.oracle_hsm_sideband`, 19

`pacifica.archiveinterface.backends.oracle_hsm_sideband.archive`, 19

`pacifica.archiveinterface.backends.oracle_hsm_sideband.extended_file_factory`, 19

`pacifica.archiveinterface.backends.oracle_hsm_sideband.orm`, 19

`pacifica.archiveinterface.backends.oracle_hsm_sideband.status`, 22

`pacifica.archiveinterface.backends.posix`, 22

`pacifica.archiveinterface.backends.posix.archive`, 22

`pacifica.archiveinterface.backends.posix.extendedfile`, 23

`pacifica.archiveinterface.backends.posix.status`, 23

`pacifica.archiveinterface.config`, 24

`pacifica.archiveinterface.exception`, 24

`pacifica.archiveinterface.globals`, 24

`pacifica.archiveinterface.id2filename`, 24

`pacifica.archiveinterface.rest_generator`, 25

`pacifica.archiveinterface.wsgi`, 26





**B**

BaseModel (class in pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.BaseModel), 19  
 bytes\_per\_level (pacifica.archiveinterface.backends.abstract.status.AbstractStatus attribute), 16  
 bytes\_type () (in module pacifica.archiveinterface.archive\_utils), 26

**C**

close () (pacifica.archiveinterface.backends.abstract.archive.AbstractBackendArchive method), 15  
 close () (pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive method), 17  
 close () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive method), 19  
 close () (pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive method), 22

**E**

copy (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamArchive attribute), 20  
 create\_time (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.SamArchive attribute), 20  
 create\_time (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode attribute), 21  
 csum (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamArchive attribute), 21  
 ctime (pacifica.archiveinterface.backends.abstract.status.AbstractStatus attribute), 16

**D**

database\_close () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.BaseModel class method), 20  
 database\_connect () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.BaseModel class method), 20  
 define\_levels () (pacifica.archiveinterface.backends.abstract.status.AbstractStatus method), 16  
 define\_levels () (pacifica.archiveinterface.backends.hpss.status.HpssStatus method), 18  
 define\_levels () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.status.HsmSidebandStatus method), 22  
 define\_levels () (pacifica.archiveinterface.backends.posix.status.PosixStatus method), 23  
 defined\_levels (pacifica.archiveinterface.backends.abstract.status.AbstractStatus attribute), 16  
 DELETE () (pacifica.archiveinterface.rest\_generator.ArchiveInterfaceGenerator method), 25

find\_file\_storage\_media() (pacifica.archiveinterface.backends.posix.status.PosixStatus method), 23

**G**

gen (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 20

gen (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

gen (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

gen (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamPath attribute), 21

GET () (pacifica.archiveinterface.rest\_generator.ArchiveInterfaceGenerator method), 25

get\_backend\_archive() (pacifica.archiveinterface.backends.factory.ArchiveBackendFactory method), 24

get\_config() (in module pacifica.archiveinterface.config), 24

get\_http\_modified\_time() (in module pacifica.archiveinterface.archive\_utils), 26

gid (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode attribute), 21

**H**

HEAD () (pacifica.archiveinterface.rest\_generator.ArchiveInterfaceGenerator method), 25

HpssBackendArchive (class in pacifica.archiveinterface.backends.hpss.archive), 17

HpssExtended (class in pacifica.archiveinterface.backends.hpss.extended), 18

HpssStatus (class in pacifica.archiveinterface.backends.hpss.status), 18

HsmSidebandBackendArchive (class in pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive), 19

HsmSidebandStatus (class in pacifica.archiveinterface.backends.oracle\_hsm\_sideband.status), 22

**I**

id (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 20

id (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 22

id2filename() (in module pacifica.archiveinterface.id2filename), 25

ino (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 20

ino (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode attribute), 21

ino (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamPath attribute), 21

**L**

load\_backend\_archive() (pacifica.archiveinterface.backends.factory.ArchiveBackendFactory method), 24

**M**

media\_type (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 20

modify\_time (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

mtime (pacifica.archiveinterface.backends.abstract.status.AbstractStatus attribute), 17

**N**

name (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

name\_hash (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

**O**

offset (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 20

online (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

open () (pacifica.archiveinterface.backends.abstract.archive.AbstractBackendArchive method), 15

open () (pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive method), 17

open () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive method), 19

open () (pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive method), 22

**P**

p\_gen (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamFile attribute), 21

pacifica.archiveinterface.backends.factory.ArchiveBackendFactory (module), 24

pacifica.archiveinterface.archive\_utils (module), 26

pacifica.archiveinterface.backends (module), 15



SamInode (class in paci- method), 19  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 21 fica.archiveinterface.backends.posix.archive.PosixBackendArchive  
 SamPath (class in paci- method), 23  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamPath (paci-  
 21 fica.archiveinterface.backends.factory.ArchiveBackendFactory  
 SamVersion (class in paci- attribute), 24  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamVersion (paci-  
 21 attribute), 20  
 seek () (pacifica.archiveinterface.backends.abstract.archive.AbstractBackendArchive (paci-  
 method), 16 attribute), 21  
 seek () (pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive (paci-  
 method), 17 attribute), 21  
 seek () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 method), 19 attribute), 21  
 seek () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamPath (paci-  
 method), 17 attribute), 21  
 seek () (pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive (paci-  
 method), 23 attribute), 21  
 seq (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 attribute), 20 method), 19  
 set\_file\_permissions () (paci- stage () (pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive (paci-  
 fica.archiveinterface.backends.abstract.archive.AbstractBackendArchive (paci-  
 method), 16 attribute), 21  
 set\_file\_permissions () (paci- attribute), 20  
 fica.archiveinterface.backends.hpss.archive.HpssBackendArchive (paci-  
 method), 17 attribute), 21  
 set\_file\_permissions () (paci- status () (pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive (paci-  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive (paci-  
 method), 19 attribute), 21  
 set\_file\_permissions () (paci- status () (pacifica.archiveinterface.backends.hpss.extended.HpssExtendedBackendArchive (paci-  
 fica.archiveinterface.backends.posix.archive.PosixBackendArchive (paci-  
 method), 23 attribute), 21  
 set\_filepath () (paci- status () (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive (paci-  
 fica.archiveinterface.backends.abstract.status.AbstractStatus (paci-  
 method), 17 attribute), 21  
 set\_filepath () (paci- T  
 fica.archiveinterface.backends.hpss.status.HpssStatus (paci-  
 method), 18 attribute), 21  
 set\_filepath () (paci- u  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandStatus (paci-  
 method), 22 attribute), 21  
 set\_filepath () (paci- uid (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 fica.archiveinterface.backends.posix.status.PosixStatus (paci-  
 method), 23 attribute), 21  
 set\_mod\_time () (paci- ut\_abs\_path () (in module paci-  
 fica.archiveinterface.backends.abstract.archive.AbstractBackendArchive (paci-  
 method), 16 attribute), 26  
 set\_mod\_time () (paci- version (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 fica.archiveinterface.backends.hpss.archive.HpssBackendArchive (paci-  
 method), 17 attribute), 22  
 set\_mod\_time () (paci- vsn (pacifica.archiveinterface.backends.oracle\_hsm\_sideband.orm.SamInode (paci-  
 fica.archiveinterface.backends.hpss.extended.HpssExtendedBackendArchive (paci-  
 method), 18 attribute), 20  
 set\_mod\_time () (paci- write () (pacifica.archiveinterface.backends.abstract.archive.AbstractBackendArchive (paci-  
 fica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive (paci-  
 method), 16 attribute), 21

`write()` (*pacifica.archiveinterface.backends.hpss.archive.HpssBackendArchive*  
*method*), 17

`write()` (*pacifica.archiveinterface.backends.oracle\_hsm\_sideband.archive.HsmSidebandBackendArchive*  
*method*), 19

`write()` (*pacifica.archiveinterface.backends.posix.archive.PosixBackendArchive*  
*method*), 23