

## Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [38. MacPython OSA Modules](#) »

## 38.2. aetools — OSA client support

Platforms: Mac

The `aetools` module contains the basic functionality on which Python AppleScript client support is built. It also imports and re-exports the core functionality of the `aetypes` and `aepack` modules. The stub packages generated by `gensuitemodule` import the relevant portions of `aetools`, so usually you do not need to import it yourself. The exception to this is when you cannot use a generated suite package and need lower-level access to scripting.

The `aetools` module itself uses the AppleEvent support provided by the `Carbon.AE` module. This has one drawback: you need access to the window manager, see section [Running scripts with a GUI](#) for details. This restriction may be lifted in future releases.

### Note

This module has been removed in Python 3.x.

The `aetools` module defines the following functions:

`aetools.packevent(ae, parameters, attributes)`

Stores parameters and attributes in a pre-created `Carbon.AE.AEDesc` object. `parameters` and `attributes` are dictionaries mapping 4-character OSA parameter keys to Python objects. The objects are packed using `aepack.pack()`.

`aetools.unpackevent(ae[, formodulename])`

Recursively unpacks a `Carbon.AE.AEDesc` event to Python objects. The function returns the parameter dictionary and the attribute dictionary. The `formodulename` argument is used by generated stub packages to control where AppleScript classes are looked up.

`aetools.keysubst(arguments, keydict)`

Converts a Python keyword argument dictionary `arguments` to the format required by `packevent` by replacing the keys, which are Python identifiers, by the four-character OSA keys according to the mapping specified in `keydict`. Used by the generated suite packages.

`aetools.enumsbst(arguments, key, edict)`

If the `arguments` dictionary contains an entry for `key` convert the value for that entry according to dictionary `edict`. This converts human-readable Python enumeration names to the OSA 4-character codes. Used by the generated suite packages.

The `aetools` module defines the following class:

`class aetools.TalkTo([signature=None, start=0, timeout=0])`

Base class for the proxy used to talk to an application. `signature` overrides the class attribute `_signature` (which is usually set by subclasses) and is the 4-char creator code defining the application to talk to. `start` can be set to true to enable running the application on class instantiation. `timeout` can be specified to change the default timeout used while waiting for an AppleEvent reply.

`TalkTo._start()`

Test whether the application is running, and attempt to start it if not.

`TalkTo.send(code, subcode[, parameters, attributes])`

Create the AppleEvent `Carbon.AE.AEDesc` for the verb with the OSA designation code, `subcode` (which are the usual 4-character strings), pack the `parameters` and `attributes` into it, send it to the target application, wait for the reply, unpack the reply with `unpackevent` and return the reply appleevent, the unpacked return values as a dictionary and the return attributes.

### Previous topic

[38.1. gensuitemodule — Generate OSA stub packages](#)

### Next topic

[38.3. aepack — Conversion between Python variables and AppleEvent data containers](#)

### This Page

- [Show Source](#)

### Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [38. MacPython OSA Modules](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.