

# Supporting Cyclic Garbage Collection

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 1)

Unknown directive type "highlight".

```
.. highlight:: c
```

Python's support for detecting and collecting garbage which involves circular references requires support from object types which are "containers" for other objects which may also be containers. Types which do not store references to other objects, or which only store references to atomic types (such as numbers or strings), do not need to provide any explicit support for garbage collection.

To create a container type, the `c:member:~PyObject.tp_flags` field of the type object must include the `const:Py_TPFLAGS_HAVE_GC` and provide an implementation of the `c:member:~PyObject.tp_traverse` handler. If instances of the type are mutable, a `c:member:~PyObject.tp_clear` implementation must also be provided.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 15); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 15); [backlink](#)

Unknown interpreted text role "const".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 15); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 15); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 21)

Unknown directive type "data".

```
.. data:: Py_TPFLAGS_HAVE_GC
   :noindex:
```

Objects with a type with this flag set must conform with the rules documented here. For convenience these objects will be referred to as container objects.

Constructors for container types must conform to two rules:

1. The memory for the object must be allocated using `c:func:PyObject_GC_New` or `c:func:PyObject_GC_NewVar`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 30); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 30); [backlink](#)

Unknown interpreted text role "c:func".

2. Once all the fields which may contain references to other containers are initialized, it must call `:c:func:'PyObject_GC_Track'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 33); [backlink](#)

Unknown interpreted text role "c:func".

Similarly, the deallocator for the object must conform to a similar pair of rules:

1. Before fields which refer to other containers are invalidated, `:c:func:'PyObject_GC_UnTrack'` must be called.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 39); [backlink](#)

Unknown interpreted text role "c:func".

2. The object's memory must be deallocated using `:c:func:'PyObject_GC_Del'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 42); [backlink](#)

Unknown interpreted text role "c:func".

### Warning

If a type adds the `Py_TPFLAGS_HAVE_GC`, then it *must* implement at least a `:c:member:'~PyObject.tp_traverse'` handler or explicitly use one from its subclass or subclasses.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 45); [backlink](#)

Unknown interpreted text role "c:member".

When calling `:c:func:'PyType_Ready'` or some of the APIs that indirectly call it like `:c:func:'PyType_FromSpecWithBases'` or `:c:func:'PyType_FromSpec'` the interpreter will automatically populate the `:c:member:'~PyObject.tp_flags'`, `:c:member:'~PyObject.tp_traverse'` and `:c:member:'~PyObject.tp_clear'` fields if the type inherits from a class that implements the garbage collector protocol and the child class does *not* include the `:const:'Py_TPFLAGS_HAVE_GC'` flag.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)**

Unknown interpreted text role "c:member".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)**

Unknown interpreted text role "c:member".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 49); [backlink](#)**

Unknown interpreted text role "const".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 57)**

Unknown directive type "c:function".

```
.. c:function:: TYPE* PyObject_GC_New(TYPE, PyTypeObject *type)
```

Analogous to :c:func:`PyObject\_New` but for container objects with the :const:`Py\_TPFLAGS\_HAVE\_GC` flag set.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 63)**

Unknown directive type "c:function".

```
.. c:function:: TYPE* PyObject_GC_NewVar(TYPE, PyTypeObject *type, Py_ssize_t size)
```

Analogous to :c:func:`PyObject\_NewVar` but for container objects with the :const:`Py\_TPFLAGS\_HAVE\_GC` flag set.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 69)**

Unknown directive type "c:function".

```
.. c:function:: TYPE* PyObject_GC_Resize(TYPE, PyVarObject *op, Py_ssize_t newsize)
```

Resize an object allocated by :c:func:`PyObject\_NewVar`. Returns the resized object or ``NULL`` on failure. \*op\* must not be tracked by the collector yet.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api) gcsupport.rst, line 75)**

Unknown directive type "c:function".

```
.. c:function:: void PyObject_GC_Track(PyObject *op)
```

Adds the object \*op\* to the set of container objects tracked by the collector. The collector can run at unexpected times so objects must be valid while being tracked. This should be called once all the fields followed by the :c:member:`~PyTypeObject.tp\_traverse` handler become valid, usually near the end of the constructor.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 84)**

Unknown directive type "c:function".

```
.. c:function:: int PyObject_IS_GC(PyObject *obj)
```

Returns non-zero if the object implements the garbage collector protocol, otherwise returns 0.

The object cannot be tracked by the garbage collector if this function returns 0.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 92)**

Unknown directive type "c:function".

```
.. c:function:: int PyObject_GC_IsTracked(PyObject *op)
```

Returns 1 if the object type of \*op\* implements the GC protocol and \*op\* is being currently tracked by the garbage collector and 0 otherwise.

This is analogous to the Python function :func:`gc.is\_tracked`.

```
.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 102)**

Unknown directive type "c:function".

```
.. c:function:: int PyObject_GC_IsFinalized(PyObject *op)
```

Returns 1 if the object type of \*op\* implements the GC protocol and \*op\* has been already finalized by the garbage collector and 0 otherwise.

This is analogous to the Python function :func:`gc.is\_finalized`.

```
.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 112)**

Unknown directive type "c:function".

```
.. c:function:: void PyObject_GC_Del(void *op)
```

Releases memory allocated to an object using :c:func:`PyObject\_GC\_New` or :c:func:`PyObject\_GC\_NewVar`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 118)**

Unknown directive type "c:function".

```
.. c:function:: void PyObject_GC_UnTrack(void *op)
```

Remove the object \*op\* from the set of container objects tracked by the collector. Note that :c:func:`PyObject\_GC\_Track` can be called again on this object to add it back to the set of tracked objects. The deallocator (:c:member:`~PyTypeObject.tp\_dealloc` handler) should call this for the object before any of the fields used by the :c:member:`~PyTypeObject.tp\_traverse` handler become invalid.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 127)**

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8
```

The `:c:func:`_PyObject_GC_TRACK`` and `:c:func:`_PyObject_GC_UNTRACK`` macros have been removed from the public C API.

The `:c:member:`~PyObject.tp_traverse`` handler accepts a function parameter of this type:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 132); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 135)

Unknown directive type "c:type".

```
.. c:type:: int (*visitproc)(PyObject *object, void *arg)
```

Type of the visitor function passed to the `:c:member:`~PyObject.tp_traverse`` handler. The function should be called with an object to traverse as `*object*` and the third parameter to the `:c:member:`~PyObject.tp_traverse`` handler as `*arg*`. The Python core uses several visitor functions to implement cyclic garbage detection; it's not expected that users will need to write their own visitor functions.

The `:c:member:`~PyObject.tp_traverse`` handler must have the following type:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 144); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 147)

Unknown directive type "c:type".

```
.. c:type:: int (*traverseproc)(PyObject *self, visitproc visit, void *arg)
```

Traversal function for a container object. Implementations must call the `*visit*` function for each object directly contained by `*self*`, with the parameters to `*visit*` being the contained object and the `*arg*` value passed to the handler. The `*visit*` function must not be called with a ```NULL``` object argument. If `*visit*` returns a non-zero value that value should be returned immediately.

To simplify writing `:c:member:`~PyObject.tp_traverse`` handlers, a `:c:func:`Py_VISIT`` macro is provided. In order to use this macro, the `:c:member:`~PyObject.tp_traverse`` implementation must name its arguments exactly `visit` and `arg`:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 156); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 156); [backlink](#)

Unknown interpreted text role "c:func".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 156); [backlink](#)

Unknown interpreted text role "c:member".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 161)**

Unknown directive type "c:function".

```
.. c:function:: void Py_VISIT(PyObject *o)

If *o* is not ``NULL``, call the *visit* callback, with arguments *o*
and *arg*. If *visit* returns a non-zero value, then return it.
Using this macro, :c:member:`~PyTypeObject.tp_traverse` handlers
look like::

    static int
    my_traverse(Noddy *self, visitproc visit, void *arg)
    {
        Py_VISIT(self->foo);
        Py_VISIT(self->bar);
        return 0;
    }
```

The :c:member:`~PyTypeObject.tp\_clear` handler must be of the :c:type:`inquiry` type, or NULL if the object is immutable.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 176); [backlink](#)**

Unknown interpreted text role "c:member".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 176); [backlink](#)**

Unknown interpreted text role "c:type".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 180)**

Unknown directive type "c:type".

```
.. c:type:: int (*inquiry) (PyObject *self)

Drop references that may have created reference cycles. Immutable objects
do not have to define this method since they can never directly create
reference cycles. Note that the object must still be valid after calling
this method (don't just call :c:func:`Py_DECREF` on a reference). The
collector will call this method if it detects that this object is involved
in a reference cycle.
```

## Controlling the Garbage Collector State

The C-API provides the following functions for controlling garbage collection runs.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 196)**

Unknown directive type "c:function".

```
.. c:function:: Py_ssize_t PyGC_Collect(void)

Perform a full garbage collection, if the garbage collector is enabled.
(Note that :func:`gc.collect` runs it unconditionally.)

Returns the number of collected + unreachable objects which cannot
be collected.
If the garbage collector is disabled or already collecting,
returns ``0`` immediately.
Errors during garbage collection are passed to :data:`sys.unraisablehook`.
This function does not raise exceptions.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-**

**main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 209)**

Unknown directive type "c:function".

```
.. c:function:: int PyGC_Enable(void)
```

Enable the garbage collector: similar to :func:`gc.enable`.  
Returns the previous state, 0 for disabled and 1 for enabled.

```
.. versionadded:: 3.10
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 217)**

Unknown directive type "c:function".

```
.. c:function:: int PyGC_Disable(void)
```

Disable the garbage collector: similar to :func:`gc.disable`.  
Returns the previous state, 0 for disabled and 1 for enabled.

```
.. versionadded:: 3.10
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api) gcsupport.rst, line 225)**

Unknown directive type "c:function".

```
.. c:function:: int PyGC_IsEnabled(void)
```

Query the state of the garbage collector: similar to :func:`gc.isenabled`.  
Returns the current state, 0 for disabled and 1 for enabled.

```
.. versionadded:: 3.10
```