

# Buffer Protocol

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

Unknown directive type "highlight".

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

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

Unknown directive type "index".

```
.. index::
    single: buffer protocol
    single: buffer interface; (see buffer protocol)
    single: buffer object; (see buffer protocol)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 13)**

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Greg Stein <gstein@lyra.org>
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 14)**

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Benjamin Peterson
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 15)**

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Stefan Krah
```

Certain objects available in Python wrap access to an underlying memory array or *buffer*. Such objects include the built-in `class`: `bytes` and `class`: `bytearray`, and some extension types like `class`: `array.array`. Third-party libraries may define their own types for special purposes, such as image processing or numeric analysis.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 18); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 18); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 18); [backlink](#)**

Unknown interpreted text role "class".

While each of these types have their own semantics, they share the common characteristic of being backed by a possibly large memory buffer. It is then desirable, in some situations, to access that buffer directly and without intermediate copying.

Python provides such a facility at the C level in the form of the `ref`: `buffer protocol <bufferobjects>`. This protocol has two sides:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 29); [backlink](#)**

Unknown interpreted text role "ref".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\cpython-main) (Doc) (c-api)buffer.rst, line 32)**

Unknown directive type "index".

```
.. index:: single: PyBufferProcs
```

- on the producer side, a type can export a "buffer interface" which allows objects of that type to expose information about their underlying buffer. This interface is described in the section `ref`: `buffer-structs`;

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

**resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 34);**  
[backlink](#)

Unknown interpreted text role "ref".

- on the consumer side, several means are available to obtain a pointer to the raw underlying data of an object (for example a method parameter).

Simple objects such as `:class:'bytes'` and `:class:'bytearray'` expose their underlying buffer in byte-oriented form. Other forms are possible; for example, the elements exposed by an `:class:'array.array'` can be multi-byte values.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 41);** [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 41);** [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 41);** [backlink](#)

Unknown interpreted text role "class".

An example consumer of the buffer interface is the `:meth:'~io.BufferedIOBase.write'` method of file objects: any object that can export a series of bytes through the buffer interface can be written to a file. While `:meth:'write'` only needs read-only access to the internal contents of the object passed to it, other methods such as `:meth:'~io.BufferedIOBase.readinto'` need write access to the contents of their argument. The buffer interface allows objects to selectively allow or reject exporting of read-write and read-only buffers.

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

Unknown interpreted text role "meth".

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

Unknown interpreted text role "meth".

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

Unknown interpreted text role "meth".

There are two ways for a consumer of the buffer interface to acquire a buffer over a target object:

- call `:c:func:'PyObject_GetBuffer'` with the right parameters;

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 56);**  
[backlink](#)

Unknown interpreted text role "c:func".

- call `:c:func:'PyArg_ParseTuple'` (or one of its siblings) with one of the `y*`, `w*` or `s*` `:ref:'format codes <arg-parsing>'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 58);**  
[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)buffer.rst, line 58);**  
[backlink](#)

Unknown interpreted text role "ref".

In both cases, `:c:func:'PyBuffer_Release'` must be called when the buffer isn't needed anymore. Failure to do so could lead to various issues such as resource leaks.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 61);** [backlink](#)

Unknown interpreted text role "c:func".

## Buffer structure

Buffer structures (or simply "buffers") are useful as a way to expose the binary data from another object to the Python programmer. They can also be used as a zero-copy slicing mechanism. Using their ability to reference a block of memory, it is possible to expose any data to the Python programmer quite easily. The memory could be a large, constant array in a C extension, it could be a raw block of memory for manipulation before passing to an operating system library, or it could be used to pass around structured data in

its native, in-memory format.

Contrary to most data types exposed by the Python interpreter, buffers are not `:type:'PyObject'` pointers but rather simple C structures. This allows them to be created and copied very simply. When a generic wrapper around a buffer is needed, a `ref:memoryview <memoryview-objects>` object can be created.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 80); [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)buffer.rst, line 80); [backlink](#)**

Unknown interpreted text role "ref".

For short instructions how to write an exporting object, see `ref:Buffer Object Structures <buffer-structs>`. For obtaining a buffer, see `:func:'PyObject_GetBuffer'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 86); [backlink](#)**

Unknown interpreted text role "ref".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 86); [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)buffer.rst, line 90)**

Unknown directive type "c:type".

```
.. c:type:: Py_buffer

.. c:member:: void *buf

A pointer to the start of the logical structure described by the buffer
fields. This can be any location within the underlying physical memory
block of the exporter. For example, with negative :c:member:`~Py_buffer.strides`
the value may point to the end of the memory block.

For :term:`contiguous` arrays, the value points to the beginning of
the memory block.

.. c:member:: void *obj

A new reference to the exporting object. The reference is owned by
the consumer and automatically decremented and set to `NULL` by
:c:func:`PyBuffer_Release`. The field is the equivalent of the return
value of any standard C-API function.

As a special case, for *temporary* buffers that are wrapped by
:c:func:`PyMemoryView_FromBuffer` or :c:func:`PyBuffer_FillInfo`
this field is `NULL`. In general, exporting objects MUST NOT
use this scheme.

.. c:member:: Py_ssize_t len

`product(shape) * itemsize`. For contiguous arrays, this is the length
of the underlying memory block. For non-contiguous arrays, it is the length
that the logical structure would have if it were copied to a contiguous
representation.

Accessing `((char *)buf)[0] up to `((char *)buf)[len-1] is only valid
if the buffer has been obtained by a request that guarantees contiguity. In
most cases such a request will be :c:macro:`PyBUF_SIMPLE` or :c:macro:`PyBUF_WRITABLE`.

.. c:member:: int readonly

An indicator of whether the buffer is read-only. This field is controlled
by the :c:macro:`PyBUF_WRITABLE` flag.

.. c:member:: Py_ssize_t itemsize

Item size in bytes of a single element. Same as the value of :func:`struct.calcsize`
called on non-`NULL` :c:member:`~Py_buffer.format` values.

Important exception: If a consumer requests a buffer without the
:c:macro:`PyBUF_FORMAT` flag, :c:member:`~Py_buffer.format` will
be set to `NULL`, but :c:member:`~Py_buffer.itemsize` still has
the value for the original format.

If :c:member:`~Py_buffer.shape` is present, the equality
`product(shape) * itemsize == len` still holds and the consumer
can use :c:member:`~Py_buffer.itemsize` to navigate the buffer.

If :c:member:`~Py_buffer.shape` is `NULL` as a result of a :c:macro:`PyBUF_SIMPLE`
or a :c:macro:`PyBUF_WRITABLE` request, the consumer must disregard
:c:member:`~Py_buffer.itemsize` and assume `itemsize == 1`.

.. c:member:: const char *format

A *NUL* terminated string in :mod:`struct` module style syntax describing
the contents of a single item. If this is `NULL`, `"B"` (unsigned bytes)
is assumed.
```

```

This field is controlled by the :c:macro:PyBUF_FORMAT flag.

.. c:member:: int ndim

The number of dimensions the memory represents as an n-dimensional array.
If it is 0, :c:member:~Py_buffer.buf points to a single item representing
a scalar. In this case, :c:member:~Py_buffer.shape, :c:member:~Py_buffer.strides
and :c:member:~Py_buffer.suboffsets MUST be NULL.

The macro :c:macro:PyBUF_MAX_NDIM limits the maximum number of dimensions
to 64. Exporters MUST respect this limit, consumers of multi-dimensional
buffers SHOULD be able to handle up to :c:macro:PyBUF_MAX_NDIM dimensions.

.. c:member:: Py_ssize_t *shape

An array of :c:type:Py_ssize_t of length :c:member:~Py_buffer.ndim
indicating the shape of the memory as an n-dimensional array. Note that
shape[0] * ... * shape[ndim-1] * itemsize MUST be equal to
:c:member:~Py_buffer.len.

Shape values are restricted to shape[n] >= 0. The case
shape[n] == 0 requires special attention. See complex arrays
for further information.

The shape array is read-only for the consumer.

.. c:member:: Py_ssize_t *strides

An array of :c:type:Py_ssize_t of length :c:member:~Py_buffer.ndim
giving the number of bytes to skip to get to a new element in each
dimension.

Stride values can be any integer. For regular arrays, strides are
usually positive, but a consumer MUST be able to handle the case
strides[n] <= 0. See complex arrays for further information.

The strides array is read-only for the consumer.

.. c:member:: Py_ssize_t *suboffsets

An array of :c:type:Py_ssize_t of length :c:member:~Py_buffer.ndim.
If suboffsets[n] >= 0, the values stored along the nth dimension are
pointers and the suboffset value dictates how many bytes to add to each
pointer after de-referencing. A suboffset value that is negative
indicates that no de-referencing should occur (striding in a contiguous
memory block).

If all suboffsets are negative (i.e. no de-referencing is needed), then
this field must be NULL (the default value).

This type of array representation is used by the Python Imaging Library
(PIL). See complex arrays for further information how to access elements
of such an array.

The suboffsets array is read-only for the consumer.

.. c:member:: void *internal

This is for use internally by the exporting object. For example, this
might be re-cast as an integer by the exporter and used to store flags
about whether or not the shape, strides, and suboffsets arrays must be
freed when the buffer is released. The consumer MUST NOT alter this
value.

```

## Buffer request types

Buffers are usually obtained by sending a buffer request to an exporting object via `:c:func:PyObject_GetBuffer`. Since the complexity of the logical structure of the memory can vary drastically, the consumer uses the *flags* argument to specify the exact buffer type it can handle.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 223); backlink**  
Unknown interpreted text role "c:func".

All `:c:data:Py_buffer` fields are unambiguously defined by the request type.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 228); backlink**  
Unknown interpreted text role "c:data".

## request-independent fields

The following fields are not influenced by *flags* and must always be filled in with the correct values: `:c:member:~Py_buffer.obj`, `:c:member:~Py_buffer.buf`, `:c:member:~Py_buffer.len`, `:c:member:~Py_buffer.itemsize`, `:c:member:~Py_buffer.ndim`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 233); 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)buffer.rst, line 233); 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)buffer.rst, line 233); [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)buffer.rst, line 233); [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)buffer.rst, line 233); [backlink](#)

Unknown interpreted text role "c:member".

## readonly, format

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 241)

Unknown directive type "c:macro".

```
.. c:macro:: PyBUF_WRITABLE
```

Controls the :c:member:`~Py\_buffer.readonly` field. If set, the exporter MUST provide a writable buffer or else report failure. Otherwise, the exporter MAY provide either a read-only or writable buffer, but the choice MUST be consistent for all consumers.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 248)

Unknown directive type "c:macro".

```
.. c:macro:: PyBUF_FORMAT
```

Controls the :c:member:`~Py\_buffer.format` field. If set, this field MUST be filled in correctly. Otherwise, this field MUST be ``NULL``.

`:c:macro:PyBUF_WRITABLE` can be |d to any of the flags in the next section. Since `:c:macro:PyBUF_SIMPLE` is defined as 0, `:c:macro:PyBUF_WRITABLE` can be used as a stand-alone flag to request a simple writable buffer.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 254); [backlink](#)

Unknown interpreted text role "c:macro".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 254); [backlink](#)

Unknown interpreted text role "c:macro".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 254); [backlink](#)

Unknown interpreted text role "c:macro".

`:c:macro:PyBUF_FORMAT` can be |d to any of the flags except `:c:macro:PyBUF_SIMPLE`. The latter already implies format B (unsigned bytes).

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 258); [backlink](#)

Unknown interpreted text role "c:macro".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 258); [backlink](#)

Unknown interpreted text role "c:macro".

## shape, strides, suboffsets

The flags that control the logical structure of the memory are listed in decreasing order of complexity. Note that each flag contains all bits of the flags below it.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 269)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{0.35\linewidth}|l|l|l|
```

Request	shape	strides	suboffsets
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 275) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_INDIRECT</pre>	yes	yes	if needed
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 277) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_STRIDES</pre>	yes	yes	NULL
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 279) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_ND</pre>	yes	NULL	NULL
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 281) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_SIMPLE</pre>	NULL	NULL	NULL

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 284)

Unknown directive type "index".

```
.. index:: contiguous, C-contiguous, Fortran contiguous
```

## contiguity requests

C or Fortran `term` 'contiguity <contiguous>' can be explicitly requested, with and without stride information. Without stride information, the buffer must be C-contiguous.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 289); [backlink](#)

Unknown interpreted text role "term".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 293)

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{0.35\linewidth}|l|l|l|l|
```

Request	shape	strides	suboffsets	contig
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 299) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_C_CONTIGUOUS</pre>	yes	yes	NULL	C

Request	shape	strides	suboffsets	contig
<div> <b>System Message: ERROR/3</b>  (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 301)  Unknown directive type "c:macro".  <pre>.. c:macro:: PyBUF_F_CONTIGUOUS</pre> </div>	yes	yes	NULL	F
<div> <b>System Message: ERROR/3</b>  (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 303)  Unknown directive type "c:macro".  <pre>.. c:macro:: PyBUF_ANY_CONTIGUOUS</pre> </div>	yes	yes	NULL	C or F
:c:macro:'PyBUF_ND' <div> <b>System Message: ERROR/3</b>  (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 305); <i>backlink</i>  Unknown interpreted text role "c:macro". </div>	yes	NULL	NULL	C

### compound requests

All possible requests are fully defined by some combination of the flags in the previous section. For convenience, the buffer protocol provides frequently used combinations as single flags.

In the following table *U* stands for undefined contiguity. The consumer would have to call `c:func:'PyBuffer_IsContiguous'` to determine contiguity.

<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 315); <i>backlink</i> Unknown interpreted text role "c:func".
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 318) Unknown directive type "tabularcolumns". <pre>.. tabularcolumns::  p{0.35\linewidth} l l l l l l l </pre>

Request	shape	strides	suboffsets	contig	readonly	format
<div> <b>System Message: ERROR/3</b>  (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 324)  Unknown directive type "c:macro".  <pre>.. c:macro:: PyBUF_FULL</pre> </div>	yes	yes	if needed	U	0	yes
<div> <b>System Message: ERROR/3</b>  (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 326)  Unknown directive type "c:macro".  <pre>.. c:macro:: PyBUF_FULL_RO</pre> </div>	yes	yes	if needed	U	1 or 0	yes

Request	shape	strides	suboffsets	contig	readonly	format
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 328) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_RECORDS</pre>	yes	yes	NULL	U	0	yes
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 330) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_RECORDS_RO</pre>	yes	yes	NULL	U	1 or 0	yes
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 332) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_STRIDED</pre>	yes	yes	NULL	U	0	NULL
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 334) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_STRIDED_RO</pre>	yes	yes	NULL	U	1 or 0	NULL
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 336) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_CONTIG</pre>	yes	NULL	NULL	C	0	NULL
<b>System Message: ERROR/3</b> (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 338) Unknown directive type "c:macro". <pre>.. c:macro:: PyBUF_CONTIG_RO</pre>	yes	NULL	NULL	C	1 or 0	NULL

## Complex arrays

### NumPy-style: shape and strides

The logical structure of NumPy-style arrays is defined by `x.member:~Py_buffer.itemsize`, `x.member:~Py_buffer.ndim`, `x.member:~Py_buffer.shape` and `x.member:~Py_buffer.strides`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 347); [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)buffer.rst, line 347); [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)buffer.rst, line 347); [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)buffer.rst, line 347); [backlink](#)

Unknown interpreted text role "c:member".

If `ndim == 0`, the memory location pointed to by `x:member:~Py_buffer.buf` is interpreted as a scalar of size `x:member:~Py_buffer.itemsize`. In that case, both `x:member:~Py_buffer.shape` and `x:member:~Py_buffer.strides` are `NULL`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 350); [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)buffer.rst, line 350); [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)buffer.rst, line 350); [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)buffer.rst, line 350); [backlink](#)

Unknown interpreted text role "c:member".

If `x:member:~Py_buffer.strides` is `NULL`, the array is interpreted as a standard n-dimensional C-array. Otherwise, the consumer must access an n-dimensional array as follows:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 354); [backlink](#)

Unknown interpreted text role "c:member".

```
ptr = (char *)buf + indices[0] * strides[0] + ... + indices[n-1] * strides[n-1];
item = *((typeof(item) *)ptr);
```

As noted above, `x:member:~Py_buffer.buf` can point to any location within the actual memory block. An exporter can check the validity of a buffer with this function:

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 364); [backlink](#)

Unknown interpreted text role "c:member".

```
def verify_structure(memlen, itemsize, ndim, shape, strides, offset):
    """Verify that the parameters represent a valid array within
    the bounds of the allocated memory:
    char *mem: start of the physical memory block
    memlen: length of the physical memory block
    offset: (char *)buf - mem
    """
    if offset % itemsize:
        return False
    if offset < 0 or offset+itemsize > memlen:
        return False
    if any(v % itemsize for v in strides):
        return False

    if ndim <= 0:
        return ndim == 0 and not shape and not strides
    if 0 in shape:
        return True

    imin = sum(strides[j]*(shape[j]-1) for j in range(ndim)
               if strides[j] <= 0)
    imax = sum(strides[j]*(shape[j]-1) for j in range(ndim)
               if strides[j] > 0)

    return 0 <= offset+imin and offset+imax+itemsize <= memlen
```

## PIL-style: shape, strides and suboffsets

In addition to the regular items, PIL-style arrays can contain pointers that must be followed in order to get to the next element in a dimension. For example, the regular three-dimensional C-array `char v[2][2][3]` can also be viewed as an array of 2 pointers to 2 two-dimensional arrays: `char (*v[2])[2][3]`. In suboffsets representation, those two pointers can be embedded at the start of `x:member:~Py_buffer.buf`, pointing to two `char x[2][3]` arrays that can be located anywhere in memory.

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

main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 400); [backlink](#)

Unknown interpreted text role "c:member".

Here is a function that returns a pointer to the element in an N-D array pointed to by an N-dimensional index when there are both non-NULL strides and suboffsets:

```
void *get_item_pointer(int ndim, void *buf, Py_ssize_t *strides,
                      Py_ssize_t *suboffsets, Py_ssize_t *indices) {
    char *pointer = (char*)buf;
    int i;
    for (i = 0; i < ndim; i++) {
        pointer += strides[i] * indices[i];
        if (suboffsets[i] >= 0) {
            pointer = *((char**)pointer) + suboffsets[i];
        }
    }
    return (void*)pointer;
}
```

## Buffer-related functions

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 430)**

Unknown directive type "c:function".

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

Return ``1`` if *obj* supports the buffer interface otherwise ``0``. When ``1`` is
returned, it doesn't guarantee that :c:func:`PyObject_GetBuffer` will
succeed. This function always succeeds.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 437)**

Unknown directive type "c:function".

```
.. c:function:: int PyObject_GetBuffer(PyObject *exporter, Py_buffer *view, int flags)

Send a request to *exporter* to fill in *view* as specified by *flags*.
If the exporter cannot provide a buffer of the exact type, it MUST raise
:c:data:`PyExc_BufferError`, set ``view->obj`` to ``NULL`` and
return ``-1``.

On success, fill in *view*, set ``view->obj`` to a new reference
to *exporter* and return 0. In the case of chained buffer providers
that redirect requests to a single object, ``view->obj`` MAY
refer to this object instead of *exporter* (See :ref:`Buffer Object Structures <buffer-structs>`).

Successful calls to :c:func:`PyObject_GetBuffer` must be paired with calls
to :c:func:`PyBuffer_Release`, similar to :c:func:`malloc` and :c:func:`free`.
Thus, after the consumer is done with the buffer, :c:func:`PyBuffer_Release`
must be called exactly once.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 455)**

Unknown directive type "c:function".

```
.. c:function:: void PyBuffer_Release(Py_buffer *view)

Release the buffer *view* and decrement the reference count for
``view->obj``. This function MUST be called when the buffer
is no longer being used, otherwise reference leaks may occur.

It is an error to call this function on a buffer that was not obtained via
:c:func:`PyObject_GetBuffer`.
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 465)**

Unknown directive type "c:function".

```
.. c:function:: Py_ssize_t PyBuffer_SizeFromFormat(const char *format)

Return the implied :c:data:`~Py_buffer.itemsize` from :c:data:`~Py_buffer.format`.
On error, raise an exception and return -1.

.. versionadded:: 3.9
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 473)**

Unknown directive type "c:function".

```
.. c:function:: int PyBuffer_IsContiguous(const Py_buffer *view, char order)

Return ``1`` if the memory defined by the *view* is C-style (*order* is
``'C'``) or Fortran-style (*order* is ``'F'``) :term:`contiguous` or either one
```

(\*order\* is ``'A'``). Return ``0`` otherwise. This function always succeeds.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 480)**

Unknown directive type "c:function".

```
.. c:function:: void* PyBuffer_GetPointer(const Py_buffer *view, const Py_ssize_t *indices)
```

Get the memory area pointed to by the \*indices\* inside the given \*view\*. \*indices\* must point to an array of ``view->ndim`` indices.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 486)**

Unknown directive type "c:function".

```
.. c:function:: int PyBuffer_FromContiguous(const Py_buffer *view, const void *buf, Py_ssize_t len, char fort)
```

Copy contiguous \*len\* bytes from \*buf\* to \*view\*. \*fort\* can be ``'C'`` or ``'F'`` (for C-style or Fortran-style ordering). ``0`` is returned on success, ``-1`` on error.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 493)**

Unknown directive type "c:function".

```
.. c:function:: int PyBuffer_ToContiguous(void *buf, const Py_buffer *src, Py_ssize_t len, char order)
```

Copy \*len\* bytes from \*src\* to its contiguous representation in \*buf\*. \*order\* can be ``'C'`` or ``'F'`` or ``'A'`` (for C-style or Fortran-style ordering or either one). ``0`` is returned on success, ``-1`` on error.

This function fails if \*len\* != \*src->len\*.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 502)**

Unknown directive type "c:function".

```
.. c:function:: int PyObject_CopyData(Py_buffer *dest, Py_buffer *src)
```

Copy data from \*src\* to \*dest\* buffer. Can convert between C-style and or Fortran-style buffers.

``0`` is returned on success, ``-1`` on error.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 509)**

Unknown directive type "c:function".

```
.. c:function:: void PyBuffer_FillContiguousStrides(int ndims, Py_ssize_t *shape, Py_ssize_t *strides, int items)
```

Fill the \*strides\* array with byte-strides of a :term:`contiguous` (C-style if \*order\* is ``'C'`` or Fortran-style if \*order\* is ``'F'``) array of the given shape with the given number of bytes per element.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\c-api\ (cpython-main) (Doc) (c-api)buffer.rst, line 516)**

Unknown directive type "c:function".

```
.. c:function:: int PyBuffer_FillInfo(Py_buffer *view, PyObject *exporter, void *buf, Py_ssize_t len, int readonly)
```

Handle buffer requests for an exporter that wants to expose \*buf\* of size \*len\* with writability set according to \*readonly\*. \*buf\* is interpreted as a sequence of unsigned bytes.

The \*flags\* argument indicates the request type. This function always fills in \*view\* as specified by flags, unless \*buf\* has been designated as read-only and :c:macro:`PyBUF\_WRITABLE` is set in \*flags\*.

On success, set ``view->obj`` to a new reference to \*exporter\* and return 0. Otherwise, raise :c:data:`PyExc\_BufferError`, set ``view->obj`` to ``NULL`` and return ``-1``;

If this function is used as part of a :ref:`getbufferproc <buffer-structs>`, \*exporter\* MUST be set to the exporting object and \*flags\* must be passed unmodified. Otherwise, \*exporter\* MUST be ``NULL``.