```python
for i in range (0,num_of_pages-1):# add each page in pdf to pdf writer  border = [0,0,0],
    Ix1 = int(index_data.loc[i][2]-10)
    Iy1 = int(index_data.loc[i][3]-10)
    Ix2 = Ix1 + 20
    Iy2 = Iy1 + 20
    indexLink =  Link(rect=[Ix1,Iy1,Ix2,Iy2], border=[0,0,1,[3,2]], target_page_index = int(index_data.loc[i][1]-1))# links to all pages from index sheet
    print(indexLink)
    stop
    pdf_writer.add_annotation(page_number=0, annotation=indexLink)
    mylink=Link(rect=[130,60,230,25], target_page_index=0)  #return to index
    pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][2] != 'No Page':
        mylink=Link(rect=[402,436,432,356], target_page_index = int(pg_data.loc[i][2])-1)  #page to West
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][3] != 'No Page':
        mylink=Link(rect=[1188,436,1218,356], target_page_index = int(pg_data.loc[i][3])-1) #page to East
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][4] != 'No Page':
        mylink=Link(rect=[770,36,850,6], target_page_index = int(pg_data.loc[i][4])-1) #page to South
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][1] != 'No Page':
        mylink=Link(rect=[770,786,850,756],target_page_index = int(pg_data.loc[i][1])-1) #page to North
        pdf_writer.add_annotation(page_number = i+1, annotation=mylink)

with open(PDFoutput, "wb") as fp:
    pdf_writer.write(fp)
```

Fil...rer | Pr...er | Co...er

link test.py × | _markup_annotations.py ×

Python Interpreter

```
*** Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32. ***
>>>
*** Remote Interpreter Reinitialized ***
{'/Type': '/Annot', '/Subtype': '/Link', '/Rect': RectangleObject([220, 601, 240, 621]), '/Border': ['0', '0', '1', ['3', '2']], '/Dest': {'target_page_index': 1, 'fit': '/Fit', 'fit_args': []}}
Traceback (most recent call last):
  File "G:\PWENG\Cartographic\Storm\DDpage\python files\link test.py", line 34, in <module>
    stop
NameError: name 'stop' is not defined
>>>
```

THIS IS BEFORE THE FIX

```
    Ix2 = Ix1 + 20
    Iy2 = Iy1 + 20
    indexLink =  Link(rect=[Ix1,Iy1,Ix2,Iy2], border=[0,0,1,[3,2]], target_page_index = int(index_data.loc[i][1]-1))# links to all pages from index sheet
    print(indexLink)
    stop
    pdf_writer.add_annotation(page_number=0, annotation=indexLink)
    mylink=Link(rect=[130,60,230,25], target_page_index=0)  #return to index
    pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][2] != 'No Page':
        mylink=Link(rect=[402,436,432,356], target_page_index = int(pg_data.loc[i][2])-1)  #page to West
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][3] != 'No Page':
        mylink=Link(rect=[1188,436,1218,356], target_page_index = int(pg_data.loc[i][3])-1) #page to East
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][4] != 'No Page':
        mylink=Link(rect=[770,36,850,6], target_page_index = int(pg_data.loc[i][4])-1) #page to South
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][1] != 'No Page':
        mylink=Link(rect=[770,786,850,756],target_page_index = int(pg_data.loc[i][1])-1) #page to North
        pdf_writer.add_annotation(page_number = i+1, annotation=mylink)

with open(PDFoutput, "wb") as fp:
    pdf_writer.write(fp)
```

Fil...rer | Pr...er | Co...er | link test.py

Python Interpreter

```
*** Python 3.9.0 (tags/v3.9.0:9cf6752, Oct  5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32. ***
*** Remote Python engine is active ***
>>>
*** Remote Interpreter Reinitialized ***
{'/Type': '/Annot', '/Subtype': '/Link', '/Rect': RectangleObject([220, 601, 240, 621]), '/Border': [0, 0, 1, [3, 2]], '/Dest': {'target_page_index': 1, 'fit': '/Fit', 'fit_args': []}}
Traceback (most recent call last):
  File "G:\PWENG\Cartographic\Storm\DDpage\python files\link test.py", line 38, in <module>
    stop
NameError: name 'stop' is not defined
>>>
```

THIS IS WITH THE FIX

```
for i in range (0,num_of_pages-1):# add each page in pdf to pdf writer  border = [0,0,0],
    Ix1 = int(index_data.loc[i][2]-10)
    Iy1 = int(index_data.loc[i][3]-10)
    Ix2 = Ix1 + 20
    Iy2 = Iy1 + 20
    indexLink =  Link(rect=[Ix1,Iy1,Ix2,Iy2], border=[0,0,1,[3,2]], target_page_index = int(index_data.loc[i][1]-1))# links to all pages from index sheet
    #print(indexLink)
    pdf_writer.add_annotation(page_number=0, annotation=indexLink)
    mylink=Link(rect=[130,60,230,25], target_page_index=0)  #return to index
    pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][2] != 'No Page':
        mylink=Link(rect=[402,436,432,356], target_page_index = int(pg_data.loc[i][2])-1)  #page to West
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][3] != 'No Page':
        mylink=Link(rect=[1188,436,1218,356], target_page_index = int(pg_data.loc[i][3])-1) #page to East
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][4] != 'No Page':
        mylink=Link(rect=[770,36,850,6], target_page_index = int(pg_data.loc[i][4])-1) #page to South
        pdf_writer.add_annotation(page_number=i+1, annotation=mylink)
    if pg_data.loc[i][1] != 'No Page':
        mylink=Link(rect=[770,786,850,756],target_page_index = int(pg_data.loc[i][1])-1) #page to North
        pdf_writer.add_annotation(page_number = i+1, annotation=mylink)

with open(PDFoutput, "wb") as fp:
    pdf_writer.write(fp)
```

Fil...rer | Pr...er | Co...er | link test.py | _markup_annotations.py

Python Interpreter

```
Incorrect first char in NameObject:(0)
Incorrect first char in NameObject:(1)
Incorrect first char in NameObject:(3)
Incorrect first char in NameObject:(2)
Incorrect first char in NameObject:(0)
Incorrect first char in NameObject:(0)
Incorrect first char in NameObject:(1)
Incorrect first char in NameObject:(3)
Incorrect first char in NameObject:(2)
>>>
```

THIS IS THE WARNING MESAGE CREATED BY IT BEING
A NameObject

Update _markup_annotations.py #2447 #2451

Open rsinger417 wants to merge 1 commit into py-pdf:main from rsinger417:patch-2

Conversation 1 · Commits 1 · Checks 0 · Files changed 1

Changes from all commits · File filter · Conversations · Jump to · ⚙

0 / 1 files viewed

4 pypdf/annotations/_markup_annotations.py

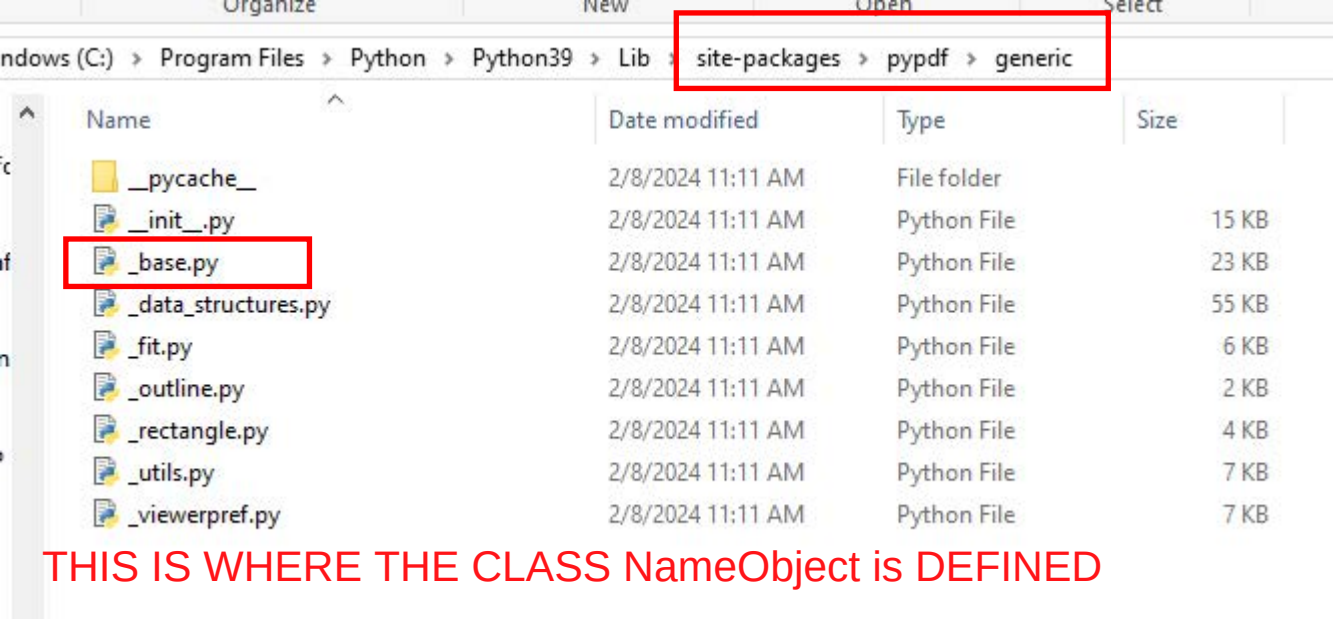
```
@@ -325,9 +325,9 @@ def __init__(
```

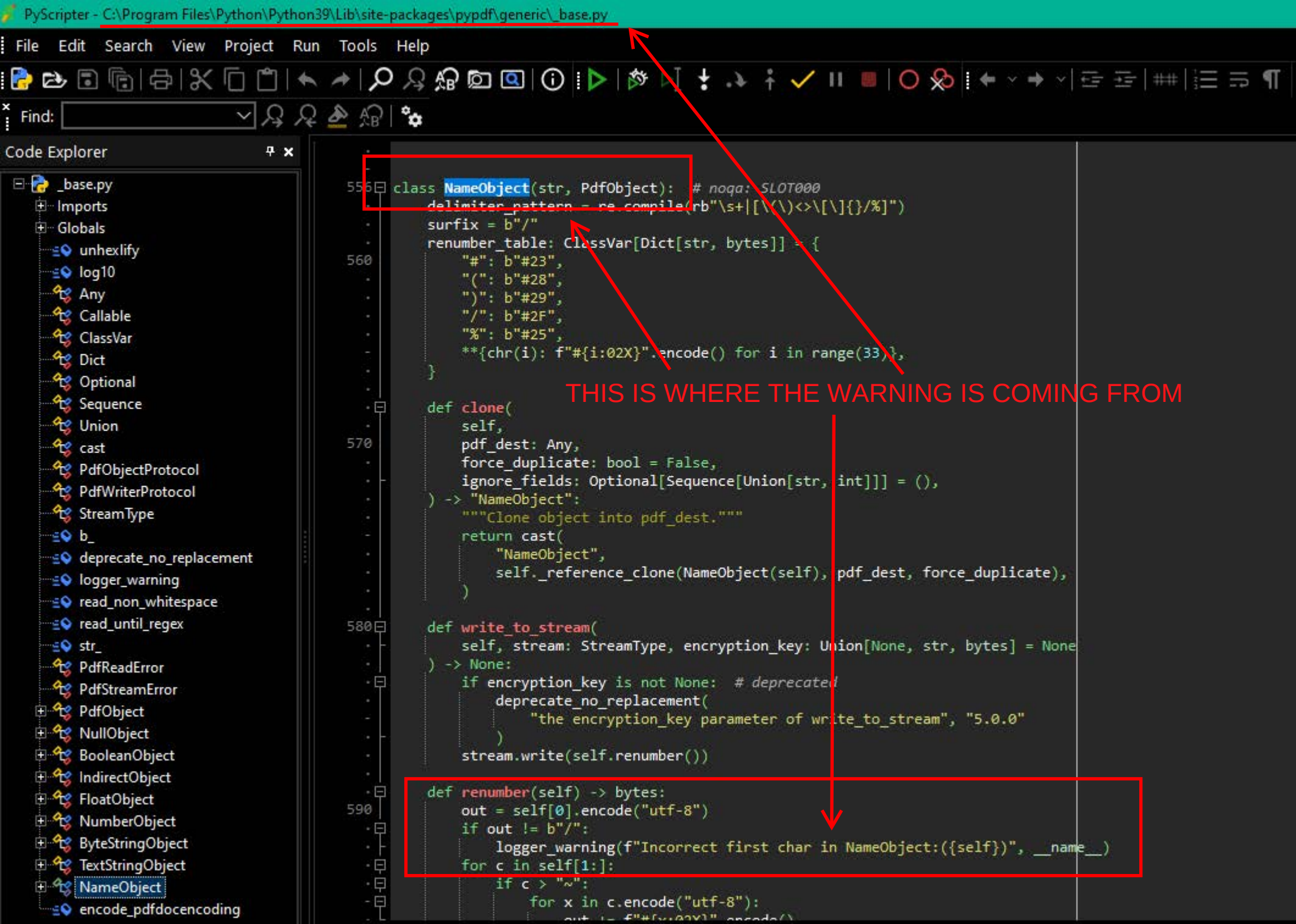
```
325
326         border_arr: BorderArrayType
327         if border is not None:
328 -           border_arr = [NameObject(n) for n in border[:3]]
329             if len(border) == 4:
330 -               dash_pattern = ArrayObject([NameObject(n) for n in border[3]])
331                 border_arr.append(dash_pattern)
332         else:
333             border_arr = [NumberObject(0)] * 3
```

```
325
326         border_arr: BorderArrayType
327         if border is not None:
328 +           border_arr = [NumberObject(n) for n in border[:3]]
329             if len(border) == 4:
330 +               dash_pattern = ArrayObject([NumberObject(n) for n in border[3]])
331                 border_arr.append(dash_pattern)
332         else:
333             border_arr = [NumberObject(0)] * 3
```

THIS IS THE FIX

THIS IS THE DEFAULT CASE, NO WARNING MESSAGE BECAUSE IT IS NOT A NamedObject


Organize
New
Open
Select
ndows (C:) › Program Files › Python › Python39 › Lib › site-packages › pypdf › generic
Name
Date modified
Type
Size
__pycache__
2/8/2024 11:11 AM
File folder
__init__.py
2/8/2024 11:11 AM
Python File
15 KB
_base.py
2/8/2024 11:11 AM
Python File
23 KB
_data_structures.py
2/8/2024 11:11 AM
Python File
55 KB
_fit.py
2/8/2024 11:11 AM
Python File
6 KB
_outline.py
2/8/2024 11:11 AM
Python File
2 KB
_rectangle.py
2/8/2024 11:11 AM
Python File
4 KB
_utils.py
2/8/2024 11:11 AM
Python File
7 KB
_viewerpref.py
2/8/2024 11:11 AM
Python File
7 KB
THIS IS WHERE THE CLASS NameObject is DEFINED


PyScripter - C:\Program Files\Python\Python39\Lib\site-packages\pypdf\generic_base.py
File Edit Search View Project Run Tools Help
Find:
Code Explorer
_base.py
Imports
Globals
unhexlify
log10
Any
Callable
ClassVar
Dict
Optional
Sequence
Union
cast
PdfObjectProtocol
PdfWriterProtocol
StreamType
b_
deprecate_no_replacement
logger_warning
read_non_whitespace
read_until_regex
str_
PdfReadError
PdfStreamError
PdfObject
NullObject
BooleanObject
IndirectObject
FloatObject
NumberObject
ByteStringObject
TextStringObject
NameObject
encode_pdfdocencoding
class NameObject(str, PdfObject): # noqa: SLOT000
delimiter_pattern = re.compile(rb"\s+|[\(\)<>\[\]{}/%]")
surfix = b"/"
renumber_table: ClassVar[Dict[str, bytes]] = {
"#": b"#23",
"(": b"#28",
")": b"#29",
"/": b"#2F",
"%": b"#25",
**{chr(i): f"#{i:02X}".encode() for i in range(33)},
}
def clone(
self,
pdf_dest: Any,
force_duplicate: bool = False,
ignore_fields: Optional[Sequence[Union[str, int]]] = (),
) -> "NameObject":
"""Clone object into pdf_dest."""
return cast(
"NameObject",
self._reference_clone(NameObject(self), pdf_dest, force_duplicate),
)
def write_to_stream(
self, stream: StreamType, encryption_key: Union[None, str, bytes] = None
) -> None:
if encryption_key is not None: # deprecated
deprecate_no_replacement(
"the encryption_key parameter of write_to_stream", "5.0.0"
)
stream.write(self.renumber())
def renumber(self) -> bytes:
out = self[0].encode("utf-8")
if out != b"/":
logger_warning(f"Incorrect first char in NameObject:({self})", __name__)
for c in self[1:]:
if c > "~":
for x in c.encode("utf-8"):
THIS IS WHERE THE WARNING IS COMING FROM

PDF 32000-1:2008

The border seem to be fine in the final pdf file with or with out the fix.
The fix will eliminate the warning meassage when run, the values should be integers not strings. The class should be NumberObject not NameObject as shown in the default case in line 333.

Table 164 – Entries common to all annotation dictionaries (continued)

Key	Type	Value
Border	array	*(Optional)* An array specifying the characteristics of the annotation's border, which shall be drawn as a rounded rectangle. *(PDF 1.0)* The array consists of three numbers defining the horizontal corner radius, vertical corner radius, and border width, all in default user space units. If the corner radii are 0, the border has square (not rounded) corners; if the border width is 0, no border is drawn. line 330 *(PDF 1.1)* The array may have a fourth element, an optional *dash array* defining a pattern of dashes and gaps that shall be used in drawing the border. The dash array shall be specified in the same format as in the line dash pattern parameter of the graphics state (see 8.4.3.6, "Line Dash Pattern"). line 328 EXAMPLE A **Border** value of [0 0 1 [3 2]] specifies a border 1 unit wide, with square corners, drawn with 3-unit dashes alternating with 2-unit gaps. NOTE *(PDF 1.2)* The dictionaries for some annotation types (such as free text and polygon annotations) can include the **BS** entry. That entry specifies a border style dictionary that has more settings than the array specified for the **Border** entry. If an annotation dictionary includes the **BS** entry, then the **Border** entry is ignored. Default value: [0 0 1]. [0, 0, 0] is pypdf default which is no border