

# libprs500

## API Documentation

November 6, 2006

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Package libprs500</b>	<b>5</b>
1.1 Modules . . . . .	5
1.2 Variables . . . . .	5
<b>2 Module libprs500.communicate</b>	<b>6</b>
2.1 Variables . . . . .	6
2.2 Class File . . . . .	6
2.2.1 Methods . . . . .	6
2.2.2 Properties . . . . .	7
2.2.3 Instance Variables . . . . .	7
2.3 Class DeviceDescriptor . . . . .	7
2.3.1 Methods . . . . .	7
2.4 Class PRS500Device . . . . .	8
2.4.1 Methods . . . . .	8
2.4.2 Properties . . . . .	10
2.4.3 Class Variables . . . . .	10
<b>3 Module libprs500.errors</b>	<b>11</b>
3.1 Class ProtocolError . . . . .	11
3.1.1 Methods . . . . .	11
3.1.2 Properties . . . . .	12
3.1.3 Class Variables . . . . .	12
3.2 Class PacketError . . . . .	13
3.2.1 Methods . . . . .	13
3.2.2 Properties . . . . .	14
3.2.3 Class Variables . . . . .	14
3.3 Class ArgumentError . . . . .	14
3.3.1 Methods . . . . .	15
3.3.2 Properties . . . . .	16
3.3.3 Class Variables . . . . .	16
3.4 Class PathError . . . . .	16
3.4.1 Methods . . . . .	16
3.4.2 Properties . . . . .	17
3.4.3 Class Variables . . . . .	18
3.5 Class ControlError . . . . .	18

3.5.1	Methods . . . . .	18
3.5.2	Properties . . . . .	19
3.5.3	Class Variables . . . . .	19
<b>4</b>	<b>Module libprs500.prstypes</b>	<b>20</b>
4.1	Variables . . . . .	20
4.2	Class TransferBuffer . . . . .	20
4.2.1	Methods . . . . .	21
4.2.2	Properties . . . . .	25
4.3	Class Command . . . . .	25
4.3.1	Methods . . . . .	25
4.3.2	Properties . . . . .	29
4.4	Class ShortCommand . . . . .	30
4.4.1	Methods . . . . .	30
4.4.2	Properties . . . . .	34
4.4.3	Class Variables . . . . .	34
4.5	Class DirOpen . . . . .	34
4.5.1	Methods . . . . .	35
4.5.2	Properties . . . . .	39
4.5.3	Class Variables . . . . .	39
4.6	Class DirRead . . . . .	39
4.6.1	Methods . . . . .	40
4.6.2	Properties . . . . .	44
4.6.3	Class Variables . . . . .	44
4.7	Class DirClose . . . . .	44
4.7.1	Methods . . . . .	45
4.7.2	Properties . . . . .	49
4.7.3	Class Variables . . . . .	49
4.8	Class LongCommand . . . . .	49
4.8.1	Methods . . . . .	50
4.8.2	Properties . . . . .	54
4.8.3	Class Variables . . . . .	54
4.9	Class AcknowledgeBulkRead . . . . .	54
4.9.1	Methods . . . . .	54
4.9.2	Properties . . . . .	58
4.9.3	Class Variables . . . . .	59
4.10	Class DeviceInfoQuery . . . . .	59
4.10.1	Methods . . . . .	59
4.10.2	Properties . . . . .	63
4.10.3	Class Variables . . . . .	63
4.11	Class FileClose . . . . .	64
4.11.1	Methods . . . . .	64
4.11.2	Properties . . . . .	68
4.11.3	Class Variables . . . . .	68
4.12	Class FileOpen . . . . .	68
4.12.1	Methods . . . . .	69
4.12.2	Properties . . . . .	73
4.12.3	Class Variables . . . . .	73
4.13	Class FileRead . . . . .	73
4.13.1	Methods . . . . .	74
4.13.2	Properties . . . . .	78
4.13.3	Class Variables . . . . .	78

4.14	Class PathQuery . . . . .	78
4.14.1	Methods . . . . .	79
4.14.2	Properties . . . . .	83
4.14.3	Class Variables . . . . .	83
4.15	Class Response . . . . .	83
4.15.1	Methods . . . . .	84
4.15.2	Properties . . . . .	88
4.15.3	Class Variables . . . . .	88
4.16	Class ListResponse . . . . .	88
4.16.1	Methods . . . . .	88
4.16.2	Properties . . . . .	92
4.16.3	Class Variables . . . . .	93
4.17	Class Answer . . . . .	93
4.17.1	Methods . . . . .	93
4.17.2	Properties . . . . .	97
4.18	Class FileProperties . . . . .	98
4.18.1	Methods . . . . .	98
4.18.2	Properties . . . . .	102
4.19	Class IdAnswer . . . . .	102
4.19.1	Methods . . . . .	103
4.19.2	Properties . . . . .	107
4.20	Class DeviceInfo . . . . .	107
4.20.1	Methods . . . . .	107
4.20.2	Properties . . . . .	111
4.21	Class ListAnswer . . . . .	112
4.21.1	Methods . . . . .	112
4.21.2	Properties . . . . .	116
<b>5</b>	<b>Module libprs500.terminfo . . . . .</b>	<b>117</b>
5.1	Class TerminalController . . . . .	117
5.1.1	Methods . . . . .	117
5.1.2	Class Variables . . . . .	117
5.2	Class ProgressBar . . . . .	118
5.2.1	Methods . . . . .	119
5.2.2	Class Variables . . . . .	119
5.2.3	Instance Variables . . . . .	119
<b>6</b>	<b>Module prs500 . . . . .</b>	<b>120</b>
6.1	Functions . . . . .	120
6.2	Variables . . . . .	120
6.3	Class FileFormatter . . . . .	120
6.3.1	Methods . . . . .	120
6.3.2	Properties . . . . .	121
<b>7</b>	<b>Module struct . . . . .</b>	<b>122</b>
7.1	Functions . . . . .	122
7.2	Variables . . . . .	123
<b>8</b>	<b>Module usb . . . . .</b>	<b>124</b>
8.1	Functions . . . . .	124
8.2	Variables . . . . .	124
8.3	Class Bus . . . . .	125

8.3.1	Methods . . . . .	125
8.3.2	Properties . . . . .	126
8.3.3	Class Variables . . . . .	126
8.4	Class Configuration . . . . .	126
8.4.1	Methods . . . . .	127
8.4.2	Properties . . . . .	127
8.4.3	Class Variables . . . . .	128
8.5	Class Device . . . . .	128
8.5.1	Methods . . . . .	128
8.5.2	Properties . . . . .	129
8.5.3	Class Variables . . . . .	129
8.6	Class DeviceHandle . . . . .	130
8.6.1	Methods . . . . .	130
8.6.2	Properties . . . . .	134
8.7	Class Endpoint . . . . .	134
8.7.1	Methods . . . . .	134
8.7.2	Properties . . . . .	135
8.7.3	Class Variables . . . . .	135
8.8	Class Interface . . . . .	135
8.8.1	Methods . . . . .	135
8.8.2	Properties . . . . .	136
8.8.3	Class Variables . . . . .	136
8.9	Class USBError . . . . .	137
8.9.1	Methods . . . . .	137
8.9.2	Properties . . . . .	138
8.9.3	Class Variables . . . . .	138

# 1 Package libprs500

This package provides an interface to the SONY Reader PRS-500 over USB.

The public interface of libprs500 is in `libprs500.communicate`. To use it

```
>>> from libprs500.communicate import PRS500Device
>>> dev = PRS500Device()
>>> dev.open()
>>> dev.get_device_information()
('Sony Reader', 'PRS-500/U', '1.0.00.21081', 'application/x-bbeb-book')
>>> dev.close()
```

There is also a script `prs500` that provides a command-line interface to libprs500. See the script for more usage examples.

The packet structure used by the SONY Reader USB protocol is defined in the module `prstypes`. The communication logic is defined in the module `communicate`.

This package requires PyUSB<sup>1</sup>. In order to use it as a non-root user on Linux, you should have the following rule in `/etc/udev/rules.d/90-local.rules` :

```
BUS=="usb", SYSFS{idProduct}=="029b", SYSFS{idVendor}=="054c", MODE="660", GROUP="plugdev"
```

You may have to adjust the GROUP and the location of the rules file to suit your distribution.

## 1.1 Modules

- **communicate**: Contains the logic for communication with the device (a SONY PRS-500).  
(Section 2, p. 6)
- **errors**: Defines the errors that libprs500 generates.  
(Section 3, p. 11)
- **prstypes**: Defines the structure of packets that are sent to/received from the device.  
(Section 4, p. 20)
- **terminfo** (Section 5, p. 117)

## 1.2 Variables

Name	Description
VERSION	<b>Value:</b> '0.1'
__author__	<b>Value:</b> 'Kovid Goyal <kovid@kovidgoyal.net>'

---

<sup>1</sup><http://pyusb.berlios.de/>

## 2 Module *libprs500.communicate*

Contains the logic for communication with the device (a SONY PRS-500).

The public interface of class `PRS500Device` defines the methods for performing various tasks.

### 2.1 Variables

Name	Description
<code>MINIMUM_COL_WIDTH</code>	Minimum width of columns in ls output <b>Value:</b> 12

### 2.2 Class File

object  **libprs500.communicate.File**

Wrapper that allows easy access to all information about files/directories

#### 2.2.1 Methods

**`__init__(self, file)`**  
`x.__init__(...)` initializes x; see `x.__class__.__doc__` for signature  
 Overrides: `object.__init__` `exitit`(inherited documentation)

**`__repr__(self)`**  
 Return path to self  
 Overrides: `object.__repr__`

**`__delattr__(...)`**  
`x.__delattr__('name') <==> del x.name`

**`__getattr__(...)`**  
`x.__getattr__('name') <==> x.name`

**`__hash__(x)`**  
`hash(x)`

**`__new__(T, S, ...)`**  
**Return Value**  
 a new object with type S, a subtype of T

<b><code>__reduce__</code></b> (...)
helper for pickle

<b><code>__reduce_ex</code></b> (...)
helper for pickle

<b><code>__setattr__</code></b> (...)
<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code>

<b><code>__str__</code></b> ( <i>x</i> )
<code>str(x)</code>

### 2.2.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>

### 2.2.3 Instance Variables

Name	Description
<code>is_dir</code>	True if self is a directory
<code>is_readonly</code>	True if self is readonly
<code>size</code>	Size in bytes of self
<code>ctime</code>	Creation time of self as a epoch
<code>wtime</code>	Creation time of self as an epoch
<code>path</code>	Path to self
<code>name</code>	Name of self

## 2.3 Class *DeviceDescriptor*

Describes a USB device.

A description is composed of the Vendor Id, Product Id and Interface Id. See the USB spec<sup>2</sup>

### 2.3.1 Methods

<b><code>__init__</code></b> ( <i>self</i> , <i>vendor_id</i> , <i>product_id</i> , <i>interface_id</i> )
---

<b><code>getDevice</code></b> ( <i>self</i> )
Return the device corresponding to the device descriptor if it is available on a USB bus. Otherwise, return None. Note that the returned device has yet to be claimed or opened.

<sup>2</sup>[http://www.usb.org/developers/docs/usb\\_20\\_05122006.zip](http://www.usb.org/developers/docs/usb_20_05122006.zip)

## 2.4 Class PRS500Device

object  **libprs500.communicate.PRS500Device**

Contains the logic for performing various tasks on the reader.

The implemented tasks are:

1. Getting information about the device
2. Getting a file from the device
3. Listing of directories. See the `list` method.

### 2.4.1 Methods

**`__init__(self, log_packets=False)`**

**Parameters**

`log_packets`: If true the packet stream to/from the device is logged

Overrides: `object.__init__`

**`open(self)`**

Claim an interface on the device for communication. Requires write privileges to the device file.

**To Do:** Check this on Mac OSX

**`close(self)`**

Release device interface

**`get_device_information(self)`**

Return (device name, device version, software version on device, mime type). See `_get_device_information`

**`get_file(self, path, outfile)`**

Read the file at path on the device and write it to outfile. For the logic see `_get_file`.

**Parameters**

`outfile`: file object like `sys.stdout` or the result of an `open` call

---

**list**(*self*, *path*, *recurse=False*)

---

Return a listing of *path*.  
 See `_list` for the communication logic.

**Parameters**

**path:** The path to list  
           (*type=string*)  
**recurse:** If true do a recursive listing  
           (*type=boolean*)

**Return Value**

A list of tuples. The first element of each tuple is a path. The second element is a list of **Files**. The path is the path we are listing, the **Files** are the files/directories in that path. If it is a recursive list, then the first element will be (**path**, children), the next will be (child, its children) and so on.

---

**\_\_delattr\_\_**(...)

---

`x.__delattr__('name')` <==> `del x.name`

---

**\_\_getattr\_\_**(...)

---

`x.__getattr__('name')` <==> `x.name`

---

**\_\_hash\_\_**(*x*)

---

`hash(x)`

---

**\_\_new\_\_**(*T*, *S*, ...)

---

**Return Value**

a new object with type *S*, a subtype of *T*

---

**\_\_reduce\_\_**(...)

---

helper for pickle

---

**\_\_reduce\_ex\_\_**(...)

---

helper for pickle

---

**\_\_repr\_\_**(*x*)

---

`repr(x)`

---

**\_\_setattr\_\_**(...)

---

`x.__setattr__('name', value)` <==> `x.name = value`

<code>__str__(x)</code>
<code>str(x)</code>

### 2.4.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>

### 2.4.3 Class Variables

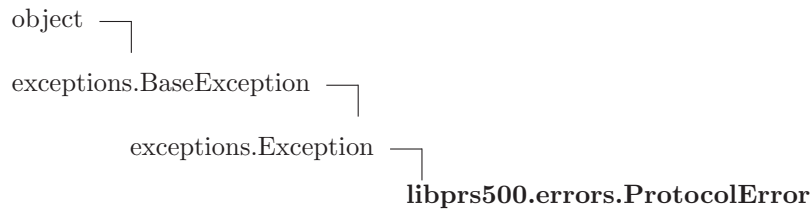
Name	Description
<code>SONY_VENDOR_ID</code>	SONY Vendor Id <b>Value:</b> 1356
<code>PRS500_PRODUCT_ID</code>	Product Id for the PRS-500 <b>Value:</b> 667
<code>PRS500_INTERFACE_ID</code>	The interface we use to talk to the device <b>Value:</b> 0
<code>PRS500_BULK_IN_EP</code>	Endpoint for Bulk reads <b>Value:</b> 129
<code>PRS500_BULK_OUT_EP</code>	Endpoint for Bulk writes <b>Value:</b> 2

## 3 Module `libprs500.errors`

Defines the errors that `libprs500` generates.

(GRAPH)

### 3.1 Class `ProtocolError`



**Known Subclasses:** `libprs500.errors.ArgumentError`, `libprs500.errors.ControlError`, `libprs500.errors.PacketError`

The base class for all exceptions in this package

#### 3.1.1 Methods

<code>__init__(self, msg)</code> <code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature Overrides: <code>exceptions.Exception.__init__</code> <code>exitit</code> (inherited documentation)
---

<code>__delattr__(...)</code> <code>x.__delattr__('name')</code> <==> <code>del x.name</code> Overrides: <code>object.__delattr__</code>
--

<code>__getattr__(...)</code> <code>x.__getattr__('name')</code> <==> <code>x.name</code> Overrides: <code>object.__getattr__</code>
--

<code>__getitem__(x, y)</code> <code>x[y]</code>
---

<code>__hash__(x)</code> <code>hash(x)</code>
--

<code>__new__(T, S, ...)</code> <b>Return Value</b> a new object with type <code>S</code> , a subtype of <code>T</code> Overrides: <code>exceptions.BaseException.__new__</code>
---

```
__reduce__(...)  
helper for pickle  
Overrides: object.__reduce__ exitit(inherited documentation)
```

```
__reduce_ex__(...)  
helper for pickle
```

```
__repr__(x)  
repr(x)  
Overrides: object.__repr__
```

```
__setattr__(...)  
x.__setattr__('name', value) <==> x.name = value  
Overrides: object.__setattr__
```

```
__setstate__(...)
```

```
__str__(x)  
str(x)  
Overrides: object.__str__
```

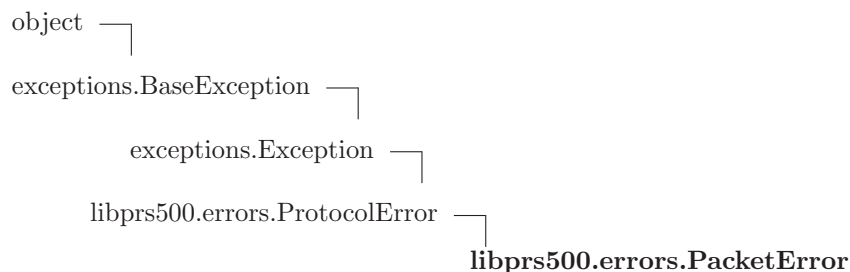
### 3.1.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>args</code>	<b>Value:</b> <attribute ' <code>args</code> ' of ' <code>exceptions.BaseException</code> '-objects>

### 3.1.3 Class Variables

Name	Description
<code>message</code>	<b>Value:</b> <member ' <code>message</code> ' of ' <code>exceptions.BaseException</code> '-objects>

## 3.2 Class `PacketError`



Errors with creating/interpreting packets

### 3.2.1 Methods

<b><code>__delattr__</code></b> (...)
<code>x.__delattr__('name') &lt;==&gt; del x.name</code>
Overrides: <code>object.__delattr__</code>

<b><code>__getattr__</code></b> (...)
<code>x.__getattr__('name') &lt;==&gt; x.name</code>
Overrides: <code>object.__getattr__</code>

<b><code>__getitem__</code></b> ( <i>x</i> , <i>y</i> )
<code>x[y]</code>

<b><code>__hash__</code></b> ( <i>x</i> )
<code>hash(x)</code>

<b><code>__init__</code></b> ( <i>self</i> , <i>msg</i> )
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>exceptions.Exception.__init__</code> <code>exitit</code> (inherited documentation)

<b><code>__new__</code></b> ( <i>T</i> , <i>S</i> , ...)
<b>Return Value</b>
a new object with type <code>S</code> , a subtype of <code>T</code>
Overrides: <code>exceptions.BaseException.__new__</code>

<b><code>__reduce__</code></b> (...)
helper for pickle
Overrides: <code>object.__reduce__</code> <code>exitit</code> (inherited documentation)

<b><code>__reduce_ex__</code></b> (...)
helper for pickle
<b><code>__repr__</code></b> ( <i>x</i> )
<code>repr(x)</code> Overrides: <code>object.__repr__</code>
<b><code>__setattr__</code></b> (...)
<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code> Overrides: <code>object.__setattr__</code>
<b><code>__setstate__</code></b> (...)
<b><code>__str__</code></b> ( <i>x</i> )
<code>str(x)</code> Overrides: <code>object.__str__</code>

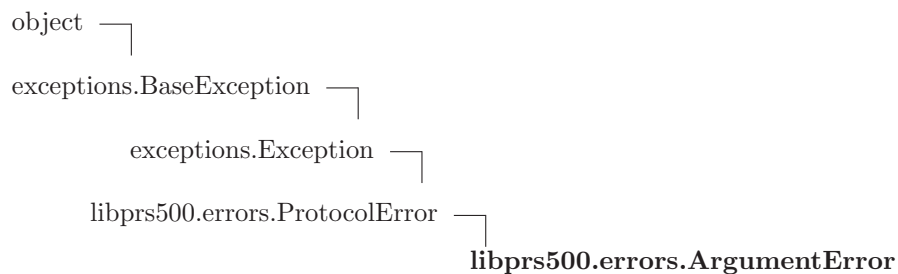
### 3.2.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>args</code>	<b>Value:</b> <attribute ' <code>args</code> ' of ' <code>exceptions.BaseException</code> '-objects>

### 3.2.3 Class Variables

Name	Description
<code>message</code>	<b>Value:</b> <member ' <code>message</code> ' of ' <code>exceptions.BaseException</code> '-objects>

## 3.3 Class *ArgumentError*



**Known Subclasses:** `libprs500.errors.PathError`

Errors caused by invalid arguments to a public interface function

### 3.3.1 Methods

**`__delattr__`**(...)

`x.__delattr__('name') <==> del x.name`

Overrides: `object.__delattr__`

**`__getattr__`**(...)

`x.__getattr__('name') <==> x.name`

Overrides: `object.__getattr__`

**`__getitem__`**(*x*, *y*)

`x[y]`

**`__hash__`**(*x*)

`hash(x)`

**`__init__`**(*self*, *msg*)

`x.__init__()` initializes `x`; see `x.__class__.__doc__` for signature

Overrides: `exceptions.Exception.__init__` `exitit`(inherited documentation)

**`__new__`**(*T*, *S*, ...)

#### Return Value

a new object with type `S`, a subtype of `T`

Overrides: `exceptions.BaseException.__new__`

**`__reduce__`**(...)

helper for pickle

Overrides: `object.__reduce__` `exitit`(inherited documentation)

**`__reduce_ex`**(...)

helper for pickle

**`__repr__`**(*x*)

`repr(x)`

Overrides: `object.__repr__`

**`__setattr__`**(...)

`x.__setattr__('name', value) <==> x.name = value`

Overrides: `object.__setattr__`

**`__setstate__`**(...)

<code>__str__(x)</code>
<code>str(x)</code>
Overrides: <code>object.__str__</code>

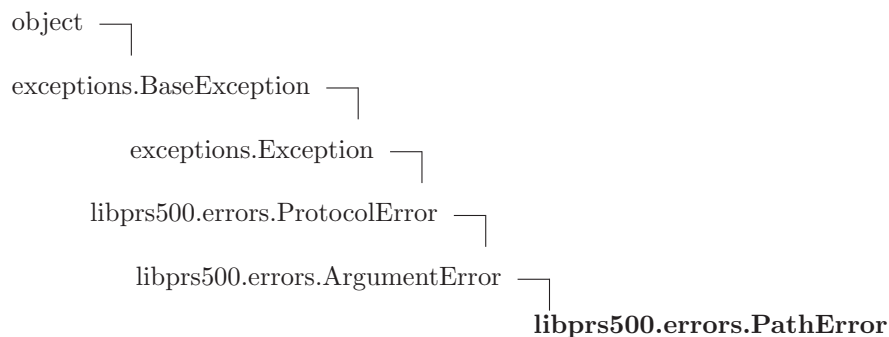
### 3.3.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>args</code>	<b>Value:</b> <attribute ' <code>args</code> ' of ' <code>exceptions.BaseException</code> '-objects>

### 3.3.3 Class Variables

Name	Description
<code>message</code>	<b>Value:</b> <member ' <code>message</code> ' of ' <code>exceptions.BaseException</code> '-objects>

## 3.4 Class *PathError*



When a user supplies an incorrect/invalid path

### 3.4.1 Methods

<code>__delattr__(...)</code>
<code>x.__delattr__('name') &lt;==&gt; del x.name</code>
Overrides: <code>object.__delattr__</code>

<code>__getattr__(...)</code>
<code>x.__getattr__('name') &lt;==&gt; x.name</code>
Overrides: <code>object.__getattr__</code>

`__getitem__(x, y)``x[y]``__hash__(x)``hash(x)``__init__(self, msg)``x.__init__(...)` initializes `x`; see `x.__class__.__doc__` for signatureOverrides: `exceptions.Exception.__init__` `exitit`(inherited documentation)`__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `exceptions.BaseException.__new__``__reduce__(...)`

helper for pickle

Overrides: `object.__reduce__` `exitit`(inherited documentation)`__reduce_ex__(...)`

helper for pickle

`__repr__(x)``repr(x)`Overrides: `object.__repr__``__setattr__(...)``x.__setattr__('name', value)`  $\iff$  `x.name = value`Overrides: `object.__setattr__``__setstate__(...)``__str__(x)``str(x)`Overrides: `object.__str__`

### 3.4.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute <code>'__class__'</code> of <code>'object'</code> objects>

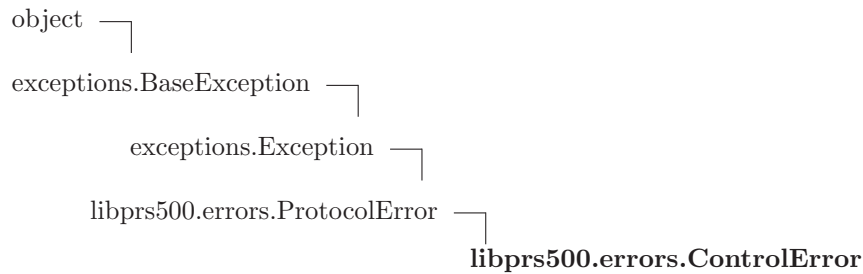
*continued on next page*

Name	Description
args	<b>Value:</b> <attribute 'args' of 'exceptions.BaseException'-objects>

### 3.4.3 Class Variables

Name	Description
message	<b>Value:</b> <member 'message' of 'exceptions.BaseException'-objects>

## 3.5 Class ControlError



Errors in Command/Response pairs while communicating with the device

### 3.5.1 Methods

```
__init__(self, query=None, response=None, desc=None)
x.__init__(...) initializes x; see x.__class__.__doc__ for signature
Overrides: libprs500.errors.ProtocolError.__init__
```

```
__str__(self)
str(x)
Overrides: exceptions.BaseException.__str__ extit(inherited documentation)
```

```
__delattr__(...)
x.__delattr__('name') <==> del x.name
Overrides: object.__delattr__
```

```
__getattr__(...)
x.__getattr__('name') <==> x.name
Overrides: object.__getattr__
```

```
__getitem__(x, y)
x[y]
```

**`__hash__`**(*x*)`hash(x)`**`__new__`**(*T*, *S*, ...)**Return Value**a new object with type *S*, a subtype of *T*Overrides: `exceptions.BaseException.__new__`**`__reduce__`**(...)

helper for pickle

Overrides: `object.__reduce__` `exitit`(inherited documentation)**`__reduce_ex__`**(...)

helper for pickle

**`__repr__`**(*x*)`repr(x)`Overrides: `object.__repr__`**`__setattr__`**(...)`x.__setattr__('name', value) <==> x.name = value`Overrides: `object.__setattr__`**`__setstate__`**(...)

### 3.5.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>args</code>	<b>Value:</b> <attribute ' <code>args</code> ' of ' <code>exceptions.BaseException</code> '-objects>

### 3.5.3 Class Variables

Name	Description
<code>message</code>	<b>Value:</b> <member ' <code>message</code> ' of ' <code>exceptions.BaseException</code> '-objects>

## 4 Module `libprs500.prstypes`

Defines the structure of packets that are sent to/received from the device.

Packet structure is defined using classes and inheritance. Each class is a view that imposes structure on the underlying data buffer. The data buffer is encoded in little-endian format, but you don't have to worry about that if you are using the classes. The classes have instance variables with getter/setter functions defined to take care of the encoding/decoding. The classes are intended to mimic C structs.

There are three kinds of packets. **Commands**, **Responses**, and **Answers**. **Commands** are sent to the device on the control bus, **Responses** are received from the device, also on the control bus. **Answers** and their sub-classes represent data packets sent to/received from the device via bulk transfers.

Commands are organized as follows: (GRAPH)

You will typically only use sub-classes of `Command`.

Responses are organized as follows: (GRAPH)

Responses inherit `Command` as they share header structure.

Answers are organized as follows: (GRAPH)

### 4.1 Variables

Name	Description
BYTE	Unsigned char little endian encoded in 1 byte <b>Value:</b> ' <code>&lt;B</code> '
WORD	Unsigned short little endian encoded in 2 bytes <b>Value:</b> ' <code>&lt;H</code> '
DWORD	Unsigned integer little endian encoded in 4 bytes <b>Value:</b> ' <code>&lt;I</code> '
DDWORD	Unsigned long long little endian encoded in 8 bytes <b>Value:</b> ' <code>&lt;Q</code> '

### 4.2 Class `TransferBuffer`



**Known Subclasses:** `libprs500.prstypes.Answer`, `libprs500.prstypes.Command`

Represents raw (unstructured) data packets sent over the usb bus.

**TransferBuffer** is a wrapper around the tuples used by PyUSB for communication. It has convenience methods to read and write data from the underlying buffer. See `TransferBuffer.pack` and `TransferBuffer.unpack`.

## 4.2.1 Methods

**\_\_init\_\_**(*self*, *packet*)Create a **TransferBuffer** from *packet* or an empty buffer.**Parameters**

**packet**: If *packet* is a list, it is copied into the **TransferBuffer** and then normalized (see **TransferBuffer.normalize**). If it is an integer, a zero buffer of that length is created.  
*(type=integer or listable object)*

Overrides: list.\_\_init\_\_

**\_\_add\_\_**(*self*, *tb*)Return a **TransferBuffer** rather than a list as the sum

Overrides: list.\_\_add\_\_

**\_\_getslice\_\_**(*self*, *start*, *end*)Return a **TransferBuffer** rather than a list as the slice

Overrides: list.\_\_getslice\_\_

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```
0700 0100 0000 0000 0000 0000 0c00 0000      .....
0200 0000 0400 0000 4461 7461                  .....Data
```

Overrides: object.\_\_str\_\_

**unpack**(*self*, *fmt*=**DWORD**, *start*=0)

Return decoded data from buffer.

**Parameters**

**fmt**: See struct<sup>a</sup>  
**start**: Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>**pack**(*self*, *val*, *fmt*=**DWORD**, *start*=0)Encode *val* and write it to buffer.**Parameters**

**fmt**: See struct<sup>a</sup>  
**start**: Position in buffer at which to write encoded data

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

**phex**(*cls*, *num*)

Return the hex representation of *num* without the 0x prefix.  
If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

**\_\_contains\_\_**(*x*, *y*)

*y* in *x*

**\_\_delattr\_\_**(...)

*x*.**\_\_delattr\_\_**('name') <==> del *x*.name

**\_\_delitem\_\_**(*x*, *y*)

del *x*[*y*]

**\_\_delslice\_\_**(*x*, *i*, *j*)

del *x*[*i*:*j*]  
Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)

*x*==*y*

**\_\_ge\_\_**(*x*, *y*)

*x*>=*y*

**\_\_getattr\_\_**(...)

*x*.**\_\_getattr\_\_**('name') <==> *x*.name  
Overrides: `object.__getattr__`

**\_\_getitem\_\_**(*x*, *y*)

*x*[*y*]

**\_\_gt\_\_**(*x*, *y*)

*x*>*y*

**\_\_hash\_\_**(*x*)

hash(*x*)  
Overrides: `object.__hash__`

`__iadd__(x, y)``x+=y``__imul__(x, y)``x*=y``__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**

a new object with type S, a subtype of T

Overrides: `object.__new__``__reduce__()`

helper for pickle

`__reduce_ex__()`

helper for pickle

`__repr__(x)``repr(x)`Overrides: `object.__repr__`

---

**`--reversed--(L)`**

return a reverse iterator over the list

---

**`--rmul--(x, n)`**

---

`n*x`

---

**`--setattr--(...)`**

---

`x.__setattr__('name', value) <==> x.name = value`

---

**`--setitem--(x, i, y)`**

---

`x[i]=y`

---

**`--setslice--(x, i, j, y)`**

---

`x[i:j]=y`

Use of negative indices is not supported.

---

**`append(L, object)`**

append object to end

---

**`count(L, value)`**

return number of occurrences of value

**Return Value****integer**

---

**`extend(L, iterable)`**

extend list by appending elements from the iterable

---

**`index(...)`**

---

`L.index(value, [start, [stop]]) -> integer` – return first index of value

---

**`insert(L, index, object)`**

insert object before index

---

**`pop(L, index=...)`**

remove and return item at index (default last)

**Return Value****item**

<b>remove</b> ( <i>L</i> , <i>value</i> )
---

remove first occurrence of value
----------------------------------

<b>reverse</b> ( <i>L</i> )
-----------------------------

reverse <i>*IN PLACE*</i>
---------------------------

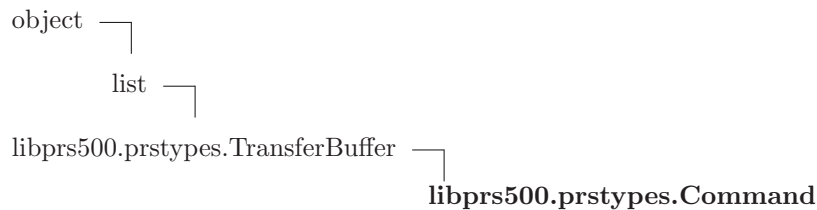
<b>sort</b> ( <i>L</i> , <i>cmp</i> =None, <i>key</i> =None, <i>reverse</i> =False)
---

stable sort <i>*IN PLACE*</i> ; cmp( <i>x</i> , <i>y</i> ) -> -1, 0, 1
--

### 4.2.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>

## 4.3 Class Command



**Known Subclasses:** `libprs500.prstypes.Response`, `libprs500.prstypes.LongCommand`, `libprs500.prstypes.DeviceInfoQuery`, `libprs500.prstypes.ShortCommand`, `libprs500.prstypes.DirOpen`, `libprs500.prstypes.FileOpen`, `libprs500.prstypes.FileRead`, `libprs500.prstypes.PathQuery`

Defines the structure of command packets sent to the device.

### 4.3.1 Methods

<b>__init__</b> ( <i>self</i> , <i>packet</i> )
---

<b>Parameters</b>
-------------------

<b>packet:</b> len(packet) > 15 or packet > 15
--

Overrides: <code>libprs500.prstypes.TransferBuffer.__init__</code>
--

<b>__add__</b> ( <i>self</i> , <i>tb</i> )
--

Return a <code>TransferBuffer</code> rather than a list as the sum
--

Overrides: <code>list.__add__</code>
--------------------------------------

<b>__contains__</b> ( <i>x</i> , <i>y</i> )
---

<i>y</i> in <i>x</i>
----------------------

---

**\_\_delattr\_\_**(...)

---

x.\_\_delattr\_\_('name') <==> del x.name

---

**\_\_delitem\_\_**(x, y)

---

del x[y]

---

**\_\_delslice\_\_**(x, i, j)

---

del x[i:j]

---

Use of negative indices is not supported.

---

**\_\_eq\_\_**(x, y)

---

x==y

---

**\_\_ge\_\_**(x, y)

---

x>=y

---

**\_\_getattr\_\_**(...)

---

x.\_\_getattr\_\_('name') <==> x.name

---

Overrides: object.\_\_getattr\_\_

---

**\_\_getitem\_\_**(x, y)

---

x[y]

---

**\_\_getslice\_\_**(self, start, end)

---

Return a TransferBuffer rather than a list as the slice

---

Overrides: list.\_\_getslice\_\_

---

**\_\_gt\_\_**(x, y)

---

x>y

---

**\_\_hash\_\_**(x)

---

hash(x)

---

Overrides: object.\_\_hash\_\_

---

**\_\_iadd\_\_**(x, y)

---

x+=y

---

**\_\_imul\_\_**(x, y)

---

x\*=y

`__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `object.__new__``__reduce__()`

helper for pickle

`__reduce_ex__()`

helper for pickle

`__repr__(x)``repr(x)`Overrides: `object.__repr__``__reversed__(L)`

return a reverse iterator over the list

`__rmul__(x, n)``n*x`

**\_\_setattr\_\_**(...)

x.\_\_setattr\_\_('name', value) &lt;==&gt; x.name = value

**\_\_setitem\_\_**(x, i, y)

x[i]=y

**\_\_setslice\_\_**(x, i, j, y)

x[i:j]=y

Use of negative indices is not supported.

**\_\_str\_\_**(self)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(L, object)

append object to end

**count**(L, value)

return number of occurrences of value

**Return Value**

integer

**extend**(L, iterable)

extend list by appending elements from the iterable

**index**(...)

L.index(value, [start, [stop]]) -&gt; integer – return first index of value

**insert**(L, index, object)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)

Encode *val* and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer at which to write encoded data

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

**phex**(*cls*, *num*)

Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**

item

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

stable sort \*IN PLACE\*; *cmp*(*x*, *y*) -> -1, 0, 1

**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters**

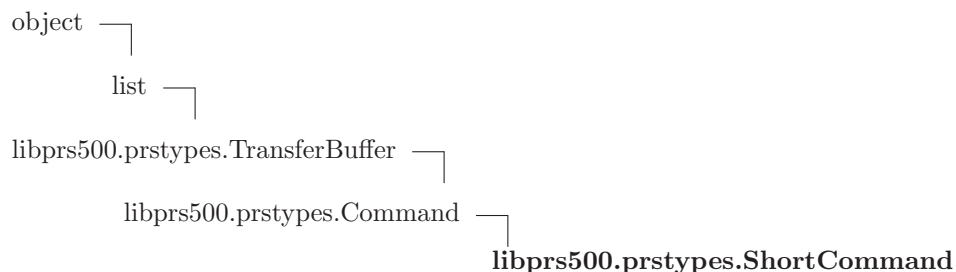
**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

### 4.3.2 Properties

Name	Description
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>

## 4.4 Class ShortCommand



**Known Subclasses:** *libprs500.prstypes.DirClose*, *libprs500.prstypes.DirRead*, *libprs500.prstypes.FileClose*

A *Command* whose data section is 4 bytes long

### 4.4.1 Methods

**`__init__(self, number=0x00, type=0x00, command=0x00)`**

**Parameters**

**number:** *Command.number*  
**type:** *Command.type*  
**command:** *ShortCommand.command*

Overrides: *libprs500.prstypes.Command.\_\_init\_\_*

**`__add__(self, tb)`**

Return a *TransferBuffer* rather than a list as the sum

Overrides: *list.\_\_add\_\_*

**`__contains__(x, y)`**

*y* in *x*

**`__delattr__(...)`**

*x.\_\_delattr\_\_('name')* <==> *del x.name*

**`__delitem__(x, y)`**

*del x[y]*

**`__delslice__(x, i, j)`**

*del x[i:j]*

Use of negative indices is not supported.

**`__eq__(x, y)`**

*x==y*

**\_\_ge\_\_**(*x*, *y*)

*x* >= *y*

**\_\_getattr\_\_**(...)

*x*.\_\_getattr\_\_('name') <==> *x*.name

Overrides: *object*.\_\_getattr\_\_

**\_\_getitem\_\_**(*x*, *y*)

*x*[*y*]

**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: *list*.\_\_getslice\_\_

**\_\_gt\_\_**(*x*, *y*)

*x* > *y*

**\_\_hash\_\_**(*x*)

hash(*x*)

Overrides: *object*.\_\_hash\_\_

**\_\_iadd\_\_**(*x*, *y*)

*x* += *y*

**\_\_imul\_\_**(*x*, *y*)

*x* \* = *y*

**\_\_iter\_\_**(*x*)

iter(*x*)

**\_\_le\_\_**(*x*, *y*)

*x* <= *y*

**\_\_len\_\_**(*x*)

len(*x*)

**\_\_lt\_\_**(*x*, *y*)

*x* < *y*

---

**\_\_mul\_\_**(*x*, *n*)

---

*x*\**n*

---

**\_\_ne\_\_**(*x*, *y*)

---

*x*!=*y*

---

**\_\_new\_\_**(*T*, *S*, ...) 

---

**Return Value**a new object with type *S*, a subtype of *T*Overrides: *object.\_\_new\_\_*

---

**\_\_reduce\_\_**(...)

---

helper for pickle

---

**\_\_reduce\_ex\_\_**(...)

---

helper for pickle

---

**\_\_repr\_\_**(*x*)

---

*repr*(*x*)Overrides: *object.\_\_repr\_\_*

---

**\_\_reversed\_\_**(*L*)

---

return a reverse iterator over the list

---

**\_\_rmul\_\_**(*x*, *n*)

---

*n*\**x*

---

**\_\_setattr\_\_**(...)

---

*x.\_\_setattr\_\_('name', value)* <==> *x.name = value*

---

**\_\_setitem\_\_**(*x*, *i*, *y*)

---

*x*[*i*]=*y*

---

**\_\_setslice\_\_**(*x*, *i*, *j*, *y*)

---

*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```

    0700 0100 0000 0000 0000 0000 0c00 0000      .....
    0200 0000 0400 0000 4461 7461                .....Data

```

Overrides: `object.__str__`**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value****integer****extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)`L.index(value, [start, [stop]])` -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=`DWORD`, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in

`TransferBuffer.__str__`**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value****item**

---

**remove**(*L*, *value*)

---

 remove first occurrence of value

---

**reverse**(*L*)

---

 reverse \*IN PLACE\*

---

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

---

 stable sort \*IN PLACE\*; cmp(*x*, *y*) -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

 Return decoded data from buffer.

**Parameters**
**fmt**: See struct<sup>a</sup>
**start**: Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

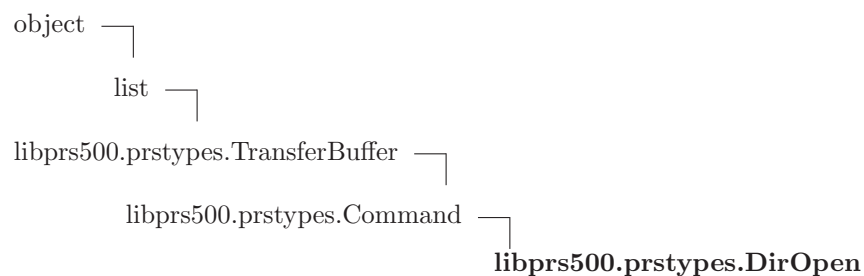
#### 4.4.2 Properties

Name	Description
command	<b>Value:</b> <property object at 0x406ae52c>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

#### 4.4.3 Class Variables

Name	Description
SIZE	Packet size in bytes <b>Value:</b> 20

### 4.5 Class DirOpen



Open a directory for reading its contents

#### 4.5.1 Methods

**`__init__(self, path)`**

Overrides: `libprs500.prstypes.Command.__init__` `exitit`(inherited documentation)

**`__add__(self, tb)`**

Return a `TransferBuffer` rather than a list as the sum

Overrides: `list.__add__`

**`__contains__(x, y)`**

`y` in `x`

**`__delattr__(...)`**

`x.__delattr__('name') <==> del x.name`

**`__delitem__(x, y)`**

`del x[y]`

**`__delslice__(x, i, j)`**

`del x[i:j]`

Use of negative indices is not supported.

**`__eq__(x, y)`**

`x==y`

**`__ge__(x, y)`**

`x>=y`

**`__getattr__(...)`**

`x.__getattr__('name') <==> x.name`

Overrides: `object.__getattr__`

**`__getitem__(x, y)`**

`x[y]`

**`__getslice__(self, start, end)`**

Return a `TransferBuffer` rather than a list as the slice

Overrides: `list.__getslice__`

`--gt--(x, y)`

`x>y`

`--hash--(x)`

`hash(x)`

Overrides: `object.__hash__`

`--iadd--(x, y)`

`x+=y`

`--imul--(x, y)`

`x*=y`

`--iter--(x)`

`iter(x)`

`--le--(x, y)`

`x<=y`

`--len--(x)`

`len(x)`

`--lt--(x, y)`

`x<y`

`--mul--(x, n)`

`x*n`

`--ne--(x, y)`

`x!=y`

`--new--(T, S, ...)`

**Return Value**

a new object with type `S`, a subtype of `T`

Overrides: `object.__new__`

`--reduce--(...)`

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000 .....

0200 0000 0400 0000 4461 7461 .....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**

integer

---

**extend**(*L, iterable*)

---

extend list by appending elements from the iterable

---

**index**(...)

---

`L.index(value, [start, [stop]])` -> integer – return first index of value

---

**insert**(*L, index, object*)

---

insert object before index

---

**pack**(*self, val, fmt=DWORD, start=0*)

---

Encode `val` and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer at which to write encoded data

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

**phex**(*cls, num*)

---

Return the hex representation of `num` without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

---

**pop**(*L, index=...*)

---

remove and return item at index (default last)

**Return Value**

item

---

**remove**(*L, value*)

---

remove first occurrence of value

---

**reverse**(*L*)

---

reverse *\*IN PLACE\**

---

**sort**(*L, cmp=None, key=None, reverse=False*)

---

stable sort *\*IN PLACE\**; `cmp(x, y)` -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

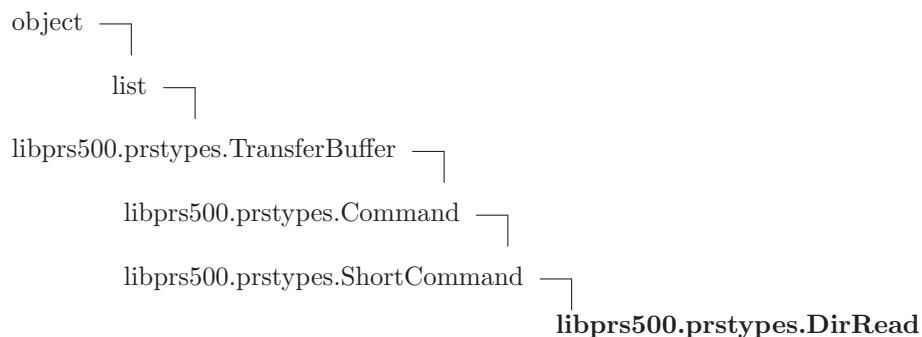
### 4.5.2 Properties

Name	Description
path_length	<b>Value:</b> <property object at 0x406ae57c>
path	<b>Value:</b> <property object at 0x406ae5a4>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

### 4.5.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 51

## 4.6 Class DirRead



The command that asks the device to send the next item in the list

### 4.6.1 Methods

**`__init__(self, id)`**

**Parameters**

**id:** The identifier returned as a result of a `DirOpen` command

Overrides: `libprs500.prstypes.ShortCommand.__init__`

**`__add__(self, tb)`**

Return a `TransferBuffer` rather than a list as the sum

Overrides: `list.__add__`

**`__contains__(x, y)`**

y in x

**`__delattr__(...)`**

x.\_\_delattr\_\_('name') <==> del x.name

**`__delitem__(x, y)`**

del x[y]

**`__delslice__(x, i, j)`**

del x[i:j]

Use of negative indices is not supported.

**`__eq__(x, y)`**

x==y

**`__ge__(x, y)`**

x>=y

**`__getattr__(...)`**

x.\_\_getattr\_\_('name') <==> x.name

Overrides: `object.__getattr__`

**`__getitem__(x, y)`**

x[y]

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object.__hash__`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--iter--`**(*x*)

`iter(x)`

**`--le--`**(*x*, *y*)

`x<=y`

**`--len--`**(*x*)

`len(x)`

**`--lt--`**(*x*, *y*)

`x<y`

**`--mul--`**(*x*, *n*)

`x*n`

**`--ne--`**(*x*, *y*)

`x!=y`

**`--new--`**(*T*, *S*, ...)

**Return Value**

    a new object with type *S*, a subtype of *T*

Overrides: `object.__new__`

**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**  
integer**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L.index*(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**  
item**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; *cmp*(x, y) -> -1, 0, 1

**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

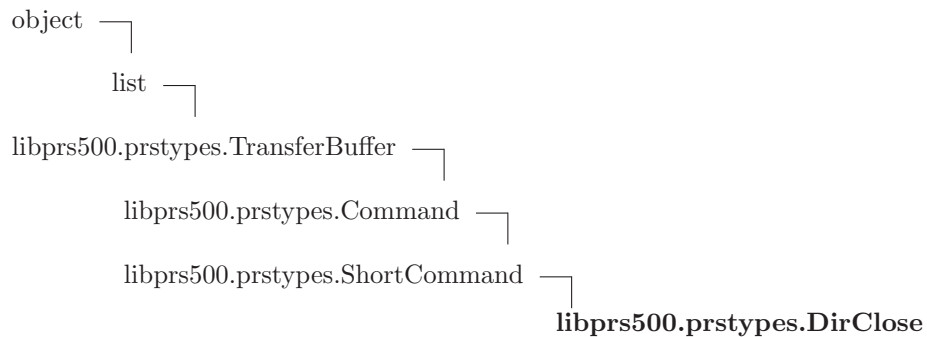
#### 4.6.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>command</code>	<b>Value:</b> <property object at 0x406ae52c>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

#### 4.6.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 53
SIZE	Packet size in bytes <b>Value:</b> 20

#### 4.7 Class *DirClose*



Close a previously opened directory

### 4.7.1 Methods

**`__init__(self, id)`**

**Parameters**

**id:** The identifier returned as a result of a `DirOpen` command

Overrides: `libprs500.prstypes.ShortCommand.__init__`

**`__add__(self, tb)`**

Return a `TransferBuffer` rather than a list as the sum

Overrides: `list.__add__`

**`__contains__(x, y)`**

y in x

**`__delattr__(...)`**

x.\_\_delattr\_\_('name') <==> del x.name

**`__delitem__(x, y)`**

del x[y]

**`__delslice__(x, i, j)`**

del x[i:j]

Use of negative indices is not supported.

**`__eq__(x, y)`**

x==y

**`__ge__(x, y)`**

x>=y

**`__getattr__(...)`**

x.\_\_getattr\_\_('name') <==> x.name

Overrides: `object.__getattr__`

**`__getitem__(x, y)`**

x[y]

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object.__hash__`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--iter--`**(*x*)

`iter(x)`

**`--le--`**(*x*, *y*)

`x<=y`

**`--len--`**(*x*)

`len(x)`

**`--lt--`**(*x*, *y*)

`x<y`

**`--mul--`**(*x*, *n*)

`x*n`

**`--ne--`**(*x*, *y*)

`x!=y`

**`--new--`**(*T*, *S*, ...)

**Return Value**

    a new object with type *S*, a subtype of *T*

Overrides: `object.__new__`

**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs, ≤ 16 bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**  
integer**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L.index*(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**  
item**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; *cmp*(x, y) -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

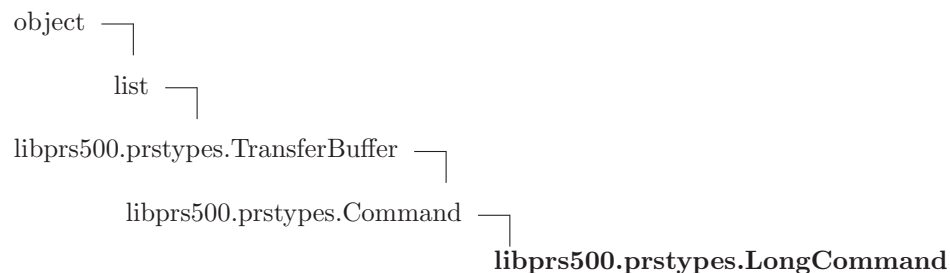
#### 4.7.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>command</code>	<b>Value:</b> <property object at 0x406ae52c>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

#### 4.7.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 52
SIZE	Packet size in bytes <b>Value:</b> 20

### 4.8 Class LongCommand



**Known Subclasses:** `libprs500.prstypes.AcknowledgeBulkRead`

A `Command` whose data section is 16 bytes long

### 4.8.1 Methods

**\_\_init\_\_**(*self*, *number*=0x00, *type*=0x00, *command*=0x00)

**Parameters**

**number:** `Command.number`  
**type:** `Command.type`  
**command:** `LongCommand.command`

Overrides: `libprs500.prstypes.Command.__init__`

**\_\_add\_\_**(*self*, *tb*)

Return a `TransferBuffer` rather than a list as the sum

Overrides: `list.__add__`

**\_\_contains\_\_**(*x*, *y*)

`y in x`

**\_\_delattr\_\_**(...)

`x.__delattr__('name') <==> del x.name`

**\_\_delitem\_\_**(*x*, *y*)

`del x[y]`

**\_\_delslice\_\_**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)

`x==y`

**\_\_ge\_\_**(*x*, *y*)

`x>=y`

**\_\_getattr\_\_**(...)

`x.__getattr__('name') <==> x.name`

Overrides: `object.__getattr__`

**\_\_getitem\_\_**(*x*, *y*)

`x[y]`

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object.__hash__`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--iter--`**(*x*)

`iter(x)`

**`--le--`**(*x*, *y*)

`x<=y`

**`--len--`**(*x*)

`len(x)`

**`--lt--`**(*x*, *y*)

`x<y`

**`--mul--`**(*x*, *n*)

`x*n`

**`--ne--`**(*x*, *y*)

`x!=y`

**`--new--`**(*T*, *S*, ...)

**Return Value**

    a new object with type *S*, a subtype of *T*

Overrides: `object.__new__`

**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**  
integer**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L.index*(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**  
item**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; *cmp*(x, y) -> -1, 0, 1

```
unpack(self, fmt=DWORD, start=0)
```

Return decoded data from buffer.

#### Parameters

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

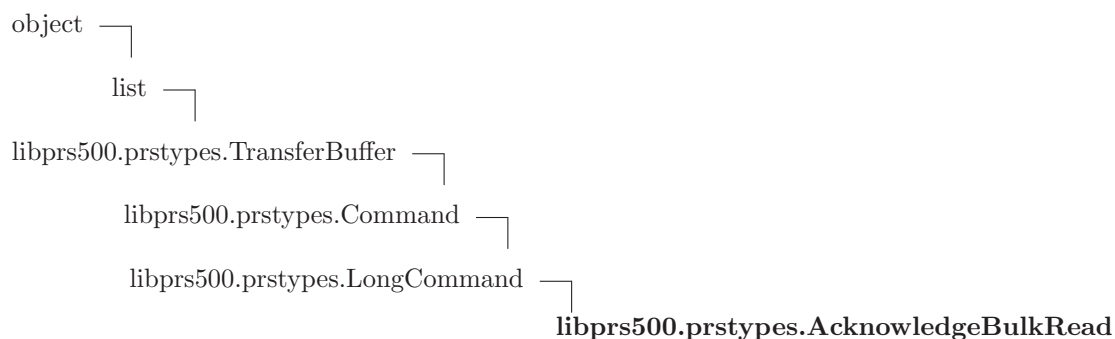
### 4.8.2 Properties

Name	Description
command	<b>Value:</b> <property object at 0x406ae66c>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

### 4.8.3 Class Variables

Name	Description
SIZE	Size in bytes of LongCommand packets <b>Value:</b> 32

## 4.9 Class *AcknowledgeBulkRead*



Must be sent to device after a bulk read

### 4.9.1 Methods

```
__init__(self, bulk_read_id)
```

bulk\_read\_id is an integer, the id of the bulk read we are acknowledging. See **Answer.id**

Overrides: libprs500.prstypes.LongCommand.\_\_init\_\_

**`__add__`**(*self*, *tb*)

Return a TransferBuffer rather than a list as the sum

Overrides: `list.__add__`

**`__contains__`**(*x*, *y*)

*y* in *x*

**`__delattr__`**(...)

*x*.`__delattr__`('name') <==> del *x*.name

**`__delitem__`**(*x*, *y*)

del *x*[*y*]

**`__delslice__`**(*x*, *i*, *j*)

del *x*[*i*:*j*]

Use of negative indices is not supported.

**`__eq__`**(*x*, *y*)

*x*==*y*

**`__ge__`**(*x*, *y*)

*x*>=*y*

**`__getattr__`**(...)

*x*.`__getattr__`('name') <==> *x*.name

Overrides: `object.__getattr__`

**`__getitem__`**(*x*, *y*)

*x*[*y*]

**`__getslice__`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`__gt__`**(*x*, *y*)

*x*>*y*

`__hash__(x)``hash(x)`Overrides: `object.__hash__``__iadd__(x, y)``x+=y``__imul__(x, y)``x*=y``__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `object.__new__``__reduce__(...)`

helper for pickle

`__reduce_ex__(...)`

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs, ≤ 16 bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000 .....

0200 0000 0400 0000 4461 7461 .....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**

integer

**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index(...)**`L.index(value, [start, [stop]])` -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**

item

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; `cmp(x, y)` -> -1, 0, 1**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer from which to decode<sup>a</sup><http://docs.python.org/lib/module-struct.html>

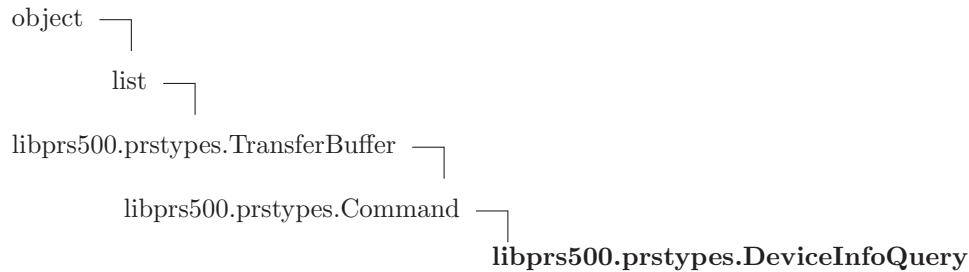
#### 4.9.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>command</code>	<b>Value:</b> <property object at 0x406ae66c>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

#### 4.9.3 Class Variables

Name	Description
<code>SIZE</code>	Size in bytes of LongCommand packets <b>Value:</b> 32

#### 4.10 Class *DeviceInfoQuery*



The command used to ask for device information

##### 4.10.1 Methods

<b><code>__init__(self)</code></b> Overrides: <code>libprs500.prstypes.Command.__init__</code> extit(inherited documentation)
<b><code>__add__(self, tb)</code></b> Return a <code>TransferBuffer</code> rather than a list as the sum Overrides: <code>list.__add__</code>
<b><code>__contains__(x, y)</code></b> y in x
<b><code>__delattr__(...)</code></b> x. <code>__delattr__</code> ('name') <==> del x.name
<b><code>__delitem__(x, y)</code></b> del x[y]

**`--delslice--`**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**`--eq--`**(*x*, *y*)

`x==y`

**`--ge--`**(*x*, *y*)

`x>=y`

**`--getattr--`**(...)

`x._getattr_('name') <==> x.name`

Overrides: `object._getattr--`

**`--getitem--`**(*x*, *y*)

`x[y]`

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list._getslice--`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object._hash--`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--iter--`**(*x*)

`iter(x)`

**`--le--`**(*x*, *y*)

`x<=y`

<code>__len__(x)</code>	<code>len(x)</code>
<code>__lt__(x, y)</code>	<code>x &lt; y</code>
<code>__mul__(x, n)</code>	<code>x * n</code>
<code>__ne__(x, y)</code>	<code>x != y</code>
<code>__new__(T, S, ...)</code> <b>Return Value</b> a new object with type S, a subtype of T Overrides: <code>object.__new__</code>	
<code>__reduce__()</code>	helper for pickle
<code>__reduce_ex__()</code>	helper for pickle
<code>__repr__(x)</code>	<code>repr(x)</code> Overrides: <code>object.__repr__</code>
<code>__reversed__(L)</code>	return a reverse iterator over the list
<code>__rmul__(x, n)</code>	<code>n * x</code>
<code>__setattr__()</code>	<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code>
<code>__setitem__(x, i, y)</code>	<code>x[i] = y</code>

---

**\_\_setslice\_\_**(*x, i, j, y*)
 

---

x[i:j]=y

Use of negative indices is not supported.

---

**\_\_str\_\_**(*self*)
 

---

Return a string representation of this buffer.

 Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```

    0700 0100 0000 0000 0000 0000 0c00 0000      .....
    0200 0000 0400 0000 4461 7461                .....Data
  
```

Overrides: object.\_\_str\_\_

---

**append**(*L, object*)
 

---

append object to end

---

**count**(*L, value*)
 

---

return number of occurrences of value

**Return Value**

integer

---

**extend**(*L, iterable*)
 

---

extend list by appending elements from the iterable

---

**index**(...)
 

---

L.index(value, [start, [stop]]) -&gt; integer – return first index of value

---

**insert**(*L, index, object*)
 

---

insert object before index

---

**pack**(*self, val, fmt=DWORD, start=0*)
 

---

 Encode **val** and write it to buffer.

**Parameters**
**fmt:** See struct<sup>a</sup>
**start:** Position in buffer at which to write encoded data

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>


---

**phex**(*cls, num*)
 

---

Return the hex representation of num without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in

TransferBuffer.\_\_str\_\_

**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**

item

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

stable sort \*IN PLACE\*; *cmp*(*x*, *y*) -> -1, 0, 1

**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters**

**fmt**: See struct<sup>a</sup>

**start**: Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

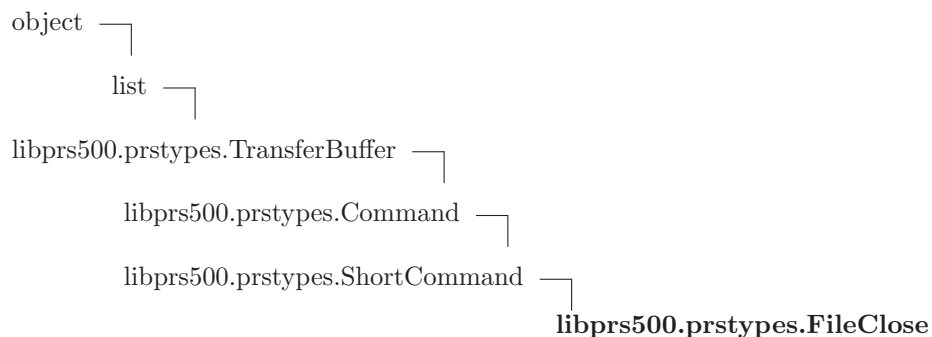
#### 4.10.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

#### 4.10.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 257

## 4.11 Class FileClose



File close command

### 4.11.1 Methods

**`__init__(self, id)`**  
 Overrides: `libprs500.prstypes.ShortCommand.__init__` `exitit`(inherited documentation)

**`__add__(self, tb)`**  
 Return a `TransferBuffer` rather than a list as the sum  
 Overrides: `list.__add__`

**`__contains__(x, y)`**  
 y in x

**`__delattr__(...)`**  
 x.`__delattr__('name')` <==> `del x.name`

**`__delitem__(x, y)`**  
`del x[y]`

**`__delslice__(x, i, j)`**  
`del x[i:j]`  
 Use of negative indices is not supported.

**`__eq__(x, y)`**  
 x==y

**`__ge__(x, y)`**  
 x>=y

**\_\_getattr\_\_**(...)`x.__getattr__('name') <==> x.name`Overrides: `object.__getattr__`**\_\_getitem\_\_**(*x*, *y*)`x[y]`**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`**\_\_gt\_\_**(*x*, *y*)`x>y`**\_\_hash\_\_**(*x*)`hash(x)`Overrides: `object.__hash__`**\_\_iadd\_\_**(*x*, *y*)`x+=y`**\_\_imul\_\_**(*x*, *y*)`x*=y`**\_\_iter\_\_**(*x*)`iter(x)`**\_\_le\_\_**(*x*, *y*)`x<=y`**\_\_len\_\_**(*x*)`len(x)`**\_\_lt\_\_**(*x*, *y*)`x<y`**\_\_mul\_\_**(*x*, *n*)`x*n`

---

**\_\_ne\_\_**(*x*, *y*)

---

*x*!=*y*

---

**\_\_new\_\_**(*T*, *S*, ...)**Return Value**a new object with type *S*, a subtype of *T*

Overrides: object.\_\_new\_\_

---

**\_\_reduce\_\_**(...)

---

helper for pickle

---

**\_\_reduce\_ex\_\_**(...)

---

helper for pickle

---

**\_\_repr\_\_**(*x*)

---

repr(*x*)

Overrides: object.\_\_repr\_\_

---

**\_\_reversed\_\_**(*L*)

---

return a reverse iterator over the list

---

**\_\_rmul\_\_**(*x*, *n*)

---

*n*\**x*

---

**\_\_setattr\_\_**(...)

---

*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value

---

**\_\_setitem\_\_**(*x*, *i*, *y*)

---

*x*[*i*]=*y*

---

**\_\_setslice\_\_**(*x*, *i*, *j*, *y*)

---

*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

---

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs, ≤ 16 bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**`integer`**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)`L.index(value, [start, [stop]])` -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=`DWORD`, *start*=0)Encode `val` and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of `num` without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**`item`**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)reverse *\*IN PLACE\**

```
sort(L, cmp=None, key=None, reverse=False)
stable sort *IN PLACE*; cmp(x, y) -> -1, 0, 1
```

```
unpack(self, fmt=DWORD, start=0)
```

Return decoded data from buffer.

#### Parameters

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

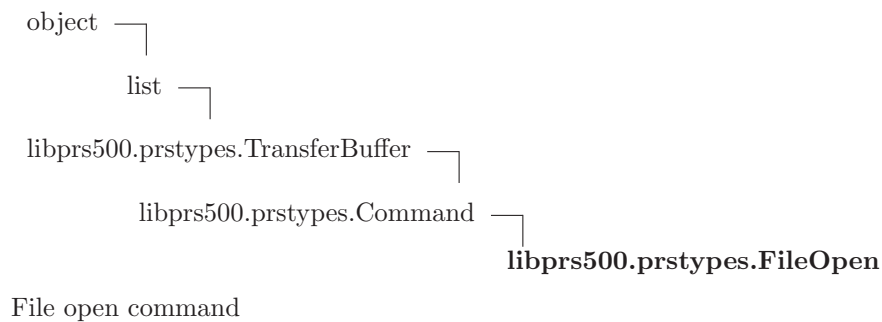
### 4.11.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>command</code>	<b>Value:</b> <property object at 0x406ae52c>
<code>data</code>	<b>Value:</b> <property object at 0x406ae4dc>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

### 4.11.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 17
SIZE	Packet size in bytes <b>Value:</b> 20

## 4.12 Class *FileOpen*



**4.12.1 Methods**

**\_\_init\_\_**(*self*, *path*, *mode=0x00*)

Overrides: *libprs500.prstypes.Command.\_\_init\_\_* *exitit*(inherited documentation)

**\_\_add\_\_**(*self*, *tb*)

Return a *TransferBuffer* rather than a list as the sum

Overrides: *list.\_\_add\_\_*

**\_\_contains\_\_**(*x*, *y*)

*y* in *x*

**\_\_delattr\_\_**(...)

*x.\_\_delattr\_\_('name')* <==> *del x.name*

**\_\_delitem\_\_**(*x*, *y*)

*del x[y]*

**\_\_delslice\_\_**(*x*, *i*, *j*)

*del x[i:j]*

Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)

*x==y*

**\_\_ge\_\_**(*x*, *y*)

*x>=y*

**\_\_getattr\_\_**(...)

*x.\_\_getattr\_\_('name')* <==> *x.name*

Overrides: *object.\_\_getattr\_\_*

**\_\_getitem\_\_**(*x*, *y*)

*x[y]*

**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a *TransferBuffer* rather than a list as the slice

Overrides: *list.\_\_getslice\_\_*

`--gt--(x, y)`

`x>y`

`--hash--(x)`

`hash(x)`

Overrides: `object.__hash__`

`--iadd--(x, y)`

`x+=y`

`--imul--(x, y)`

`x*=y`

`--iter--(x)`

`iter(x)`

`--le--(x, y)`

`x<=y`

`--len--(x)`

`len(x)`

`--lt--(x, y)`

`x<y`

`--mul--(x, n)`

`x*n`

`--ne--(x, y)`

`x!=y`

`--new--(T, S, ...)`

**Return Value**

a new object with type `S`, a subtype of `T`

Overrides: `object.__new__`

`--reduce--(...)`

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)`repr(x)`Overrides: `object.__repr__`**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)`n*x`**\_\_setattr\_\_**(...)`x.__setattr__('name', value) <==> x.name = value`**\_\_setitem\_\_**(*x*, *i*, *y*)`x[i]=y`**\_\_setslice\_\_**(*x*, *i*, *j*, *y*)`x[i:j]=y`

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
---	-------

0200 0000 0400 0000 4461 7461	.....Data
-------------------------------	-----------

Overrides: `object.__str__`**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**`integer`

---

**extend**(*L, iterable*)

---

extend list by appending elements from the iterable

---

**index**(...)

---

`L.index(value, [start, [stop]])` -> integer – return first index of value

---

**insert**(*L, index, object*)

---

insert object before index

---

**pack**(*self, val, fmt=DWORD, start=0*)

---

Encode `val` and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer at which to write encoded data

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

**phex**(*cls, num*)

---

Return the hex representation of `num` without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

---

**pop**(*L, index=...*)

---

remove and return item at index (default last)

**Return Value**

item

---

**remove**(*L, value*)

---

remove first occurrence of value

---

**reverse**(*L*)

---

reverse *\*IN PLACE\**

---

**sort**(*L, cmp=None, key=None, reverse=False*)

---

stable sort *\*IN PLACE\**; `cmp(x, y)` -> -1, 0, 1

**unpack**(self, fmt=DWORD, start=0)

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

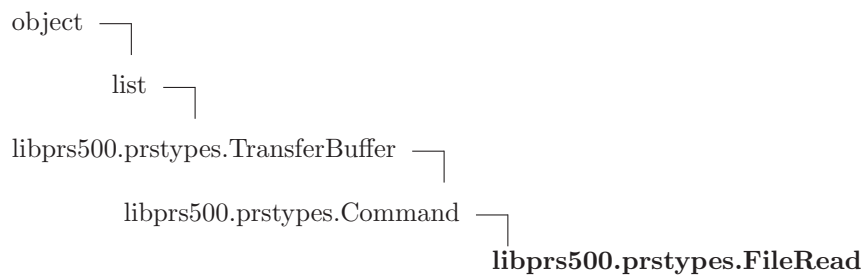
#### 4.12.2 Properties

Name	Description
mode	<b>Value:</b> <property object at 0x406ae734>
path_length	<b>Value:</b> <property object at 0x406ae75c>
path	<b>Value:</b> <property object at 0x406ae784>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

#### 4.12.3 Class Variables

Name	Description
NUMBER	<b>Value:</b> 16
READ	<b>Value:</b> 0
WRITE	<b>Value:</b> 1

#### 4.13 Class FileRead



Command to read from an open file

### 4.13.1 Methods

**`__init__`**(*self*, *id*, *offset*, *size*)

**Parameters**

**id:** File identifier returned by a `FileOpen` command  
*(type=unsigned int)*

**offset:** Position in file at which to read  
*(type=unsigned long long)*

**size:** number of bytes to read  
*(type=unsigned int)*

Overrides: `libprs500.prstypes.Command.__init__`

**`__add__`**(*self*, *tb*)

Return a `TransferBuffer` rather than a list as the sum

Overrides: `list.__add__`

**`__contains__`**(*x*, *y*)

*y* in *x*

**`__delattr__`**(...)

*x.\_\_delattr\_\_('name')* <==> `del x.name`

**`__delitem__`**(*x*, *y*)

`del x[y]`

**`__delslice__`**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**`__eq__`**(*x*, *y*)

*x*==*y*

**`__ge__`**(*x*, *y*)

*x*>=*y*

**`__getattr__`**(...)

*x.\_\_getattr\_\_('name')* <==> *x.name*

Overrides: `object.__getattr__`

**`--getitem--`**(*x*, *y*)

`x[y]`

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object.__hash__`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--iter--`**(*x*)

`iter(x)`

**`--le--`**(*x*, *y*)

`x<=y`

**`--len--`**(*x*)

`len(x)`

**`--lt--`**(*x*, *y*)

`x<y`

**`--mul--`**(*x*, *n*)

`x*n`

**`--ne--`**(*x*, *y*)

`x!=y`

**\_\_new\_\_**(*T*, *S*, ...)

**Return Value**

a new object with type *S*, a subtype of *T*

Overrides: `object.__new__`

**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)

`repr(x)`

Overrides: `object.__repr__`

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)

`n*x`

**\_\_setattr\_\_**(...)

`x.__setattr__('name', value) <==> x.name = value`

**\_\_setitem\_\_**(*x*, *i*, *y*)

`x[i]=y`

**\_\_setslice\_\_**(*x*, *i*, *j*, *y*)

`x[i:j]=y`

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```
0700 0100 0000 0000 0000 0000 0c00 0000      .....
0200 0000 0400 0000 4461 7461                  .....Data
```

Overrides: `object.__str__`

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value***integer***extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L.index*(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)

Return the hex representation of num without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value***item***remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; *cmp*(x, y) -> -1, 0, 1

```
unpack(self, fmt=DWORD, start=0)
```

Return decoded data from buffer.

#### Parameters

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

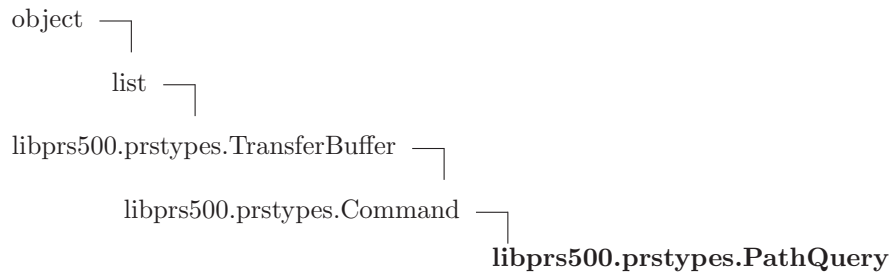
### 4.13.2 Properties

Name	Description
id	<b>Value:</b> <property object at 0x406ae7d4>
offset	<b>Value:</b> <property object at 0x406ae7fc>
size	<b>Value:</b> <property object at 0x406ae824>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

### 4.13.3 Class Variables

Name	Description
NUMBER	Command number to read from a file <b>Value:</b> 22

## 4.14 Class PathQuery



Defines structure of command that requests information about a path

```
>>> print prstypes.PathQuery("/test/path/", number=prstypes.PathQuery.PROPERTIES)
1800 0000 0100 0000 0000 0000 0f00 0000 .....
0b00 0000 2f74 6573 742f 7061 7468 2f ....test/path/
```

**4.14.1 Methods****\_\_init\_\_**(*self*, *path*)Overrides: *libprs500.prstypes.Command.\_\_init\_\_* *exitit*(inherited documentation)**\_\_add\_\_**(*self*, *tb*)

Return a TransferBuffer rather than a list as the sum

Overrides: *list.\_\_add\_\_***\_\_contains\_\_**(*x*, *y*)*y* in *x***\_\_delattr\_\_**(...)*x.\_\_delattr\_\_('name')* <==> *del x.name***\_\_delitem\_\_**(*x*, *y*)*del x[y]***\_\_delslice\_\_**(*x*, *i*, *j*)*del x[i:j]*

Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)*x==y***\_\_ge\_\_**(*x*, *y*)*x>=y***\_\_getattr\_\_**(...)*x.\_\_getattr\_\_('name')* <==> *x.name*Overrides: *object.\_\_getattr\_\_***\_\_getitem\_\_**(*x*, *y*)*x[y]***\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: *list.\_\_getslice\_\_*

---

`--gt--(x, y)`

---

`x>y`

---

`--hash--(x)`

---

`hash(x)``Overrides: object.__hash__`

---

`--iadd--(x, y)`

---

`x+=y`

---

`--imul--(x, y)`

---

`x*=y`

---

`--iter--(x)`

---

`iter(x)`

---

`--le--(x, y)`

---

`x<=y`

---

`--len--(x)`

---

`len(x)`

---

`--lt--(x, y)`

---

`x<y`

---

`--mul--(x, n)`

---

`x*n`

---

`--ne--(x, y)`

---

`x!=y`

---

`--new--(T, S, ...)`

---

**Return Value**

a new object with type `S`, a subtype of `T`

`Overrides: object.__new__`

---

`--reduce--(...)`

---

`helper for pickle`

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs, ≤ 16 bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000 .....

0200 0000 0400 0000 4461 7461 .....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**

integer

---

**extend**(*L, iterable*)

---

extend list by appending elements from the iterable

---

**index**(...)

---

`L.index(value, [start, [stop]])` -> integer – return first index of value

---

**insert**(*L, index, object*)

---

insert object before index

---

**pack**(*self, val, fmt=DWORD, start=0*)

---

Encode `val` and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer at which to write encoded data

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

**phex**(*cls, num*)

---

Return the hex representation of `num` without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

---

**pop**(*L, index=...*)

---

remove and return item at index (default last)

**Return Value**

item

---

**remove**(*L, value*)

---

remove first occurrence of value

---

**reverse**(*L*)

---

reverse *\*IN PLACE\**

---

**sort**(*L, cmp=None, key=None, reverse=False*)

---

stable sort *\*IN PLACE\**; `cmp(x, y)` -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

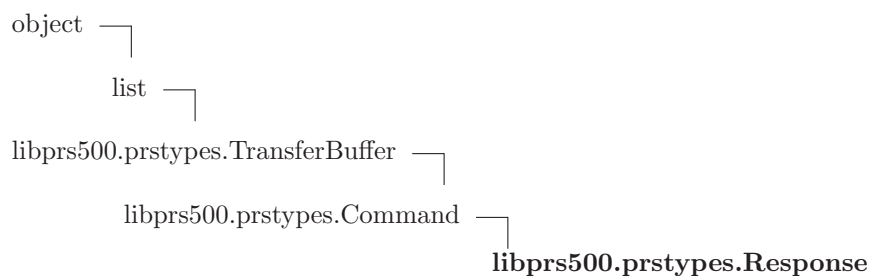
#### 4.14.2 Properties

Name	Description
path_length	<b>Value:</b> <property object at 0x406ae89c>
path	<b>Value:</b> <property object at 0x406ae8c4>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
data	<b>Value:</b> <property object at 0x406ae4dc>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

#### 4.14.3 Class Variables

Name	Description
NUMBER	Command number <b>Value:</b> 24

### 4.15 Class Response



**Known Subclasses:** libprs500.prstypes.ListResponse

Defines the structure of response packets received from the device.

**Response** inherits from **Command** as the first 16 bytes have the same structure.

**4.15.1 Methods****\_\_init\_\_**(*self*, *packet*)`len(packet) == Response.SIZE`Overrides: `libprs500.prstypes.Command.__init__`**\_\_add\_\_**(*self*, *tb*)

Return a TransferBuffer rather than a list as the sum

Overrides: `list.__add__`**\_\_contains\_\_**(*x*, *y*)`y in x`**\_\_delattr\_\_**(...)`x.__delattr__('name') <==> del x.name`**\_\_delitem\_\_**(*x*, *y*)`del x[y]`**\_\_delslice\_\_**(*x*, *i*, *j*)`del x[i:j]`

Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)`x==y`**\_\_ge\_\_**(*x*, *y*)`x>=y`**\_\_getattr\_\_**(...)`x.__getattr__('name') <==> x.name`Overrides: `object.__getattr__`**\_\_getitem\_\_**(*x*, *y*)`x[y]`**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

`--gt--(x, y)``x>y``--hash--(x)``hash(x)`Overrides: `object.__hash__``--iadd--(x, y)``x+=y``--imul--(x, y)``x*=y``--iter--(x)``iter(x)``--le--(x, y)``x<=y``--len--(x)``len(x)``--lt--(x, y)``x<y``--mul--(x, n)``x*n``--ne--(x, y)``x!=y``--new--(T, S, ...)`**Return Value**

a new object with type S, a subtype of T

Overrides: `object.__new__``--reduce--(...)`

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(x)

repr(x)

Overrides: object.\_\_repr\_\_

**\_\_reversed\_\_**(L)

return a reverse iterator over the list

**\_\_rmul\_\_**(x, n)

n\*x

**\_\_setattr\_\_**(...)

x.\_\_setattr\_\_('name', value) &lt;==&gt; x.name = value

**\_\_setitem\_\_**(x, i, y)

x[i]=y

**\_\_setslice\_\_**(x, i, j, y)

x[i:j]=y

Use of negative indices is not supported.

**\_\_str\_\_**(self)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000 .....

0200 0000 0400 0000 4461 7461 .....Data

Overrides: object.\_\_str\_\_

**append**(L, object)

append object to end

**count**(L, value)

return number of occurrences of value

**Return Value**

integer

**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)

`L.index(value, [start, [stop]])` -> integer – return first index of value

**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=`DWORD`, *start*=0)

Encode *val* and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer at which to write encoded data

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

**phex**(*cls*, *num*)

Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**

item

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse *\*IN PLACE\**

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

stable sort *\*IN PLACE\**; `cmp(x, y)` -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

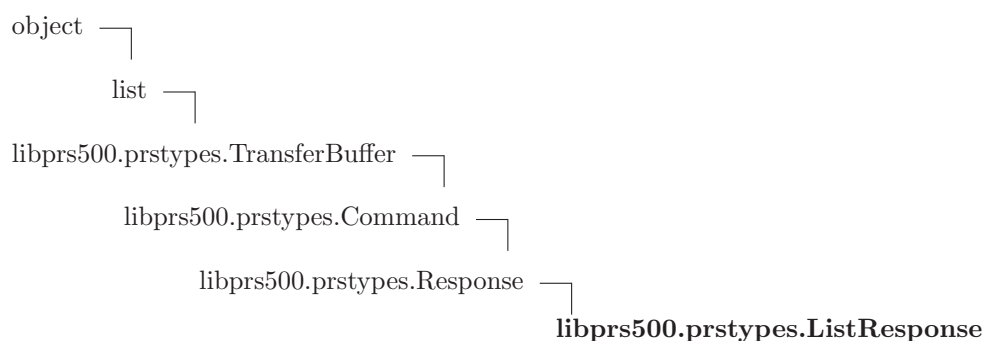
### 4.15.2 Properties

Name	Description
rnumber	<b>Value:</b> <property object at 0x406ae914>
data	<b>Value:</b> <property object at 0x406ae93c>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>
length	<b>Value:</b> <property object at 0x406ae1e4>
number	<b>Value:</b> <property object at 0x406ae144>
type	<b>Value:</b> <property object at 0x406ae194>

### 4.15.3 Class Variables

Name	Description
SIZE	Size of response packets in the SONY protocol <b>Value:</b> 32

## 4.16 Class ListResponse



Defines the structure of response packets received during list (ll) queries. See *PathQuery*.

### 4.16.1 Methods

---

**\_\_add\_\_**(*self*, *tb*)

---

Return a TransferBuffer rather than a list as the sum

Overrides: list.\_\_add\_\_

---

**`--contains--`**(*x*, *y*)

`y in x`

**`--delattr--`**(...)

`x.__delattr__('name') <==> del x.name`

**`--delitem--`**(*x*, *y*)

`del x[y]`

**`--delslice--`**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**`--eq--`**(*x*, *y*)

`x==y`

**`--ge--`**(*x*, *y*)

`x>=y`

**`--getattr--`**(...)

`x.__getattr__('name') <==> x.name`

Overrides: `object.__getattr__`

**`--getitem--`**(*x*, *y*)

`x[y]`

**`--getslice--`**(*self*, *start*, *end*)

Return a `TransferBuffer` rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object.__hash__`

**`--iadd--`**(*x*, *y*)

`x+=y`

---

`__imul__(x, y)`

---

`x*=y`

---

`__init__(self, packet)`

---

`len(packet) == Response.SIZE``Overrides: libprs500.prstypes.Command.__init__`

---

`__iter__(x)`

---

`iter(x)`

---

`__le__(x, y)`

---

`x<=y`

---

`__len__(x)`

---

`len(x)`

---

`__lt__(x, y)`

---

`x<y`

---

`__mul__(x, n)`

---

`x*n`

---

`__ne__(x, y)`

---

`x!=y`

---

`__new__(T, S, ...)`

---

**Return Value**`a new object with type S, a subtype of T``Overrides: object.__new__`

---

`__reduce__(...)`

---

`helper for pickle`

---

`__reduce_ex__(...)`

---

`helper for pickle`

---

`__repr__(x)`

---

`repr(x)``Overrides: object.__repr__`

**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n*\**x***\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*]=*y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*]=*y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs, ≤ 16 bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value****integer****extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L*.index(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)

Encode *val* and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer at which to write encoded data

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

**phex**(*cls*, *num*)

Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**

item

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

stable sort \*IN PLACE\*; *cmp*(*x*, *y*) -> -1, 0, 1

**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

#### 4.16.2 Properties

Name	Description
code	<b>Value:</b> <property object at 0x406ae98c>
is_file	<b>Value:</b> <property object at 0x406ae9b4>
is_invalid	<b>Value:</b> <property object at 0x406ae9dc>
path_not_found	<b>Value:</b> <property object at 0x406aea04>
is_unmounted	<b>Value:</b> <property object at 0x406aea2c>

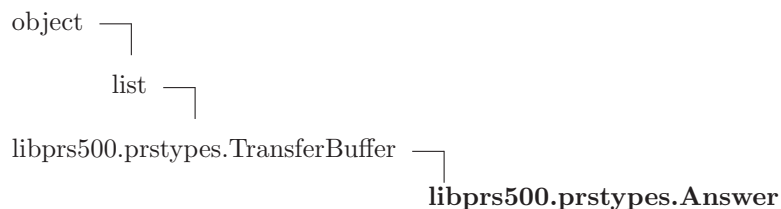
*continued on next page*

Name	Description
<code>is_eol</code>	<b>Value:</b> <property object at 0x406aea54>
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>
<code>data</code>	<b>Value:</b> <property object at 0x406ae93c>
<code>length</code>	<b>Value:</b> <property object at 0x406ae1e4>
<code>number</code>	<b>Value:</b> <property object at 0x406ae144>
<code>rnumber</code>	<b>Value:</b> <property object at 0x406ae914>
<code>type</code>	<b>Value:</b> <property object at 0x406ae194>

#### 4.16.3 Class Variables

Name	Description
<code>IS_FILE</code>	Queried path is a file <b>Value:</b> 4294967250L
<code>IS_INVALID</code>	Queried path is malformed/invalid <b>Value:</b> 4294967289L
<code>IS_UNMOUNTED</code>	Queried path is not mounted (i.e. a removed storage card/stick) <b>Value:</b> 4294967240L
<code>IS_EOL</code>	There are no more entries in the list <b>Value:</b> 4294967290L
<code>PATH_NOT_FOUND</code>	Queried path is not found <b>Value:</b> 4294967255L
<code>SIZE</code>	Size of response packets in the SONY protocol <b>Value:</b> 32

#### 4.17 Class Answer



**Known Subclasses:** `libprs500.prstypes.DeviceInfo`, `libprs500.prstypes.FileProperties`, `libprs500.prstypes.IdAnswer`, `libprs500.prstypes.ListAnswer`

Defines the structure of packets sent to host via a bulk transfer (i.e., bulk reads)

##### 4.17.1 Methods

<code>__init__(self, packet)</code>
<b>Parameters</b> <code>packet</code> : <code>len(packet) ≥ 16</code>
Overrides: <code>libprs500.prstypes.TransferBuffer.__init__</code>

**`--add--`**(*self*, *tb*)

Return a TransferBuffer rather than a list as the sum

Overrides: `list.__add__`

**`--contains--`**(*x*, *y*)

*y* in *x*

**`--delattr--`**(...)

*x*.`--delattr--`('name') <==> `del x.name`

**`--delitem--`**(*x*, *y*)

`del x[y]`

**`--delslice--`**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**`--eq--`**(*x*, *y*)

*x*==*y*

**`--ge--`**(*x*, *y*)

*x*>=*y*

**`--getattribute--`**(...)

*x*.`--getattribute--`('name') <==> *x*.*name*

Overrides: `object.__getattribute__`

**`--getitem--`**(*x*, *y*)

*x*[*y*]

**`--getslice--`**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`

**`--gt--`**(*x*, *y*)

*x*>*y*

`__hash__(x)``hash(x)`Overrides: `object.__hash__``__iadd__(x, y)``x+=y``__imul__(x, y)``x*=y``__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `object.__new__``__reduce__(...)`

helper for pickle

`__reduce_ex__(...)`

helper for pickle

**`__repr__`**(*x*)`repr(x)`Overrides: `object.__repr__`**`__reversed__`**(*L*)

return a reverse iterator over the list

**`__rmul__`**(*x*, *n*)`n*x`**`__setattr__`**(...)`x.__setattr__('name', value) <==> x.name = value`**`__setitem__`**(*x*, *i*, *y*)`x[i]=y`**`__setslice__`**(*x*, *i*, *j*, *y*)`x[i:j]=y`

Use of negative indices is not supported.

**`__str__`**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
---	-------

0200 0000 0400 0000 4461 7461	.....Data
-------------------------------	-----------

Overrides: `object.__str__`**`append`**(*L*, *object*)

append object to end

**`count`**(*L*, *value*)

return number of occurrences of value

**Return Value****integer****`extend`**(*L*, *iterable*)

extend list by appending elements from the iterable

**index(...)**`L.index(value, [start, [stop]])` -> integer – return first index of value**insert**(`L, index, object`)

insert object before index

**pack**(`self, val, fmt=DWORD, start=0`)Encode `val` and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(`cls, num`)Return the hex representation of `num` without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(`L, index=...`)

remove and return item at index (default last)

**Return Value**

item

**remove**(`L, value`)

remove first occurrence of value

**reverse**(`L`)

reverse \*IN PLACE\*

**sort**(`L, cmp=None, key=None, reverse=False`)stable sort \*IN PLACE\*; `cmp(x, y)` -> -1, 0, 1**unpack**(`self, fmt=DWORD, start=0`)

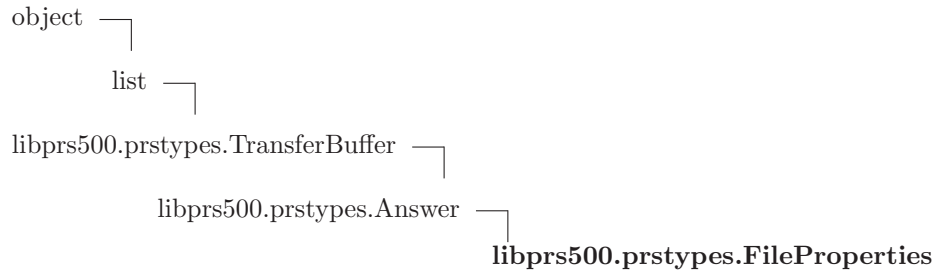
Return decoded data from buffer.

**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer from which to decode<sup>a</sup><http://docs.python.org/lib/module-struct.html>

#### 4.17.2 Properties

Name	Description
id	<b>Value:</b> <property object at 0x406aeaa4>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>

## 4.18 Class FileProperties



Defines the structure of packets that contain size, date and permissions information about files/directories.

### 4.18.1 Methods

<b>__add__</b> ( <i>self</i> , <i>tb</i> )
Return a TransferBuffer rather than a list as the sum
Overrides: list.__add__

<b>__contains__</b> ( <i>x</i> , <i>y</i> )
y in x

<b>__delattr__</b> (...)
x.__delattr__('name') <==> del x.name

<b>__delitem__</b> ( <i>x</i> , <i>y</i> )
del x[y]

<b>__delslice__</b> ( <i>x</i> , <i>i</i> , <i>j</i> )
del x[i:j]
Use of negative indices is not supported.

<b>__eq__</b> ( <i>x</i> , <i>y</i> )
x==y

<b>__ge__</b> ( <i>x</i> , <i>y</i> )
x>=y

**\_\_getattrute\_\_**(...)`x.__getattrute__('name') <==> x.name`Overrides: `object.__getattrute__`**\_\_getitem\_\_**(*x*, *y*)`x[y]`**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`**\_\_gt\_\_**(*x*, *y*)`x>y`**\_\_hash\_\_**(*x*)`hash(x)`Overrides: `object.__hash__`**\_\_iadd\_\_**(*x*, *y*)`x+=y`**\_\_imul\_\_**(*x*, *y*)`x*=y`**\_\_init\_\_**(*self*, *packet*)**Parameters**`packet: len(packet) ≥ 16`Overrides: `libprs500.prstypes.TransferBuffer.__init__`**\_\_iter\_\_**(*x*)`iter(x)`**\_\_le\_\_**(*x*, *y*)`x<=y`**\_\_len\_\_**(*x*)`len(x)`

**\_\_lt\_\_**(*x*, *y*)*x* < *y***\_\_mul\_\_**(*x*, *n*)*x* \* *n***\_\_ne\_\_**(*x*, *y*)*x* != *y***\_\_new\_\_**(*T*, *S*, ...)**Return Value**a new object with type *S*, a subtype of *T*Overrides: `object.__new__`**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)`repr(x)`Overrides: `object.__repr__`**\_\_reversed\_\_**(*L*)

return a reverse iterator over the list

**\_\_rmul\_\_**(*x*, *n*)*n* \* *x***\_\_setattr\_\_**(...)*x*.`__setattr__`('name', value) <==> *x*.name = value**\_\_setitem\_\_**(*x*, *i*, *y*)*x*[*i*] = *y***\_\_setslice\_\_**(*x*, *i*, *j*, *y*)*x*[*i*:*j*] = *y*

Use of negative indices is not supported.

**\_\_str\_\_**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```

    0700 0100 0000 0000 0000 0000 0c00 0000      .....
    0200 0000 0400 0000 4461 7461                .....Data

```

Overrides: `object.__str__`**append**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value****integer****extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)`L.index(value, [start, [stop]])` -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=`DWORD`, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in

`TransferBuffer.__str__`**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value****item**

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)reverse *\*IN PLACE\****sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort *\*IN PLACE\**; *cmp*(*x*, *y*) -> -1, 0, 1**unpack**(*self*, *fmt*=DWORD, *start*=0)

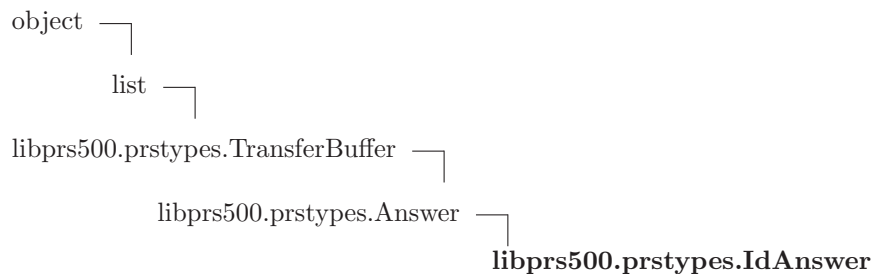
Return decoded data from buffer.

**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer from which to decode<sup>a</sup><http://docs.python.org/lib/module-struct.html>

#### 4.18.2 Properties

Name	Description
<code>file_size</code>	<b>Value:</b> <property object at 0x406aeaf4>
<code>is_dir</code>	<b>Value:</b> <property object at 0x406aeb1c>
<code>ctime</code>	<b>Value:</b> <property object at 0x406aeb44>
<code>wtime</code>	<b>Value:</b> <property object at 0x406aeb6c>
<code>is_readonly</code>	<b>Value:</b> <property object at 0x406aeb94>
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>id</code>	<b>Value:</b> <property object at 0x406aeaa4>

#### 4.19 Class `IdAnswer`



Defines the structure of packets that contain identifiers for queries.

## 4.19.1 Methods

**\_\_add\_\_**(*self*, *tb*)

Return a TransferBuffer rather than a list as the sum

Overrides: list.\_\_add\_\_

**\_\_contains\_\_**(*x*, *y*)

y in x

**\_\_delattr\_\_**(...)

x.\_\_delattr\_\_('name') &lt;==&gt; del x.name

**\_\_delitem\_\_**(*x*, *y*)

del x[y]

**\_\_delslice\_\_**(*x*, *i*, *j*)

del x[i:j]

Use of negative indices is not supported.

**\_\_eq\_\_**(*x*, *y*)

x==y

**\_\_ge\_\_**(*x*, *y*)

x&gt;=y

**\_\_getattr\_\_**(...)

x.\_\_getattr\_\_('name') &lt;==&gt; x.name

Overrides: object.\_\_getattr\_\_

**\_\_getitem\_\_**(*x*, *y*)

x[y]

**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: list.\_\_getslice\_\_

**\_\_gt\_\_**(*x*, *y*)

x&gt;y

`__hash__(x)``hash(x)`Overrides: `object.__hash__``__iadd__(x, y)``x+=y``__imul__(x, y)``x*=y``__init__(self, packet)`**Parameters**`packet: len(packet) ≥ 16`Overrides: `libprs500.prstypes.TransferBuffer.__init__``__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `object.__new__`

**`__reduce__`**(...)

helper for pickle

**`__reduce_ex`**(...)

helper for pickle

**`__repr__`**(*x*)`repr(x)`Overrides: `object.__repr__`**`__reversed__`**(*L*)

return a reverse iterator over the list

**`__rmul__`**(*x*, *n*)`n*x`**`__setattr__`**(...)`x.__setattr__('name', value) <==> x.name = value`**`__setitem__`**(*x*, *i*, *y*)`x[i]=y`**`__setslice__`**(*x*, *i*, *j*, *y*)`x[i:j]=y`

Use of negative indices is not supported.

**`__str__`**(*self*)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: `object.__str__`**`append`**(*L*, *object*)

append object to end

**count**(*L*, *value*)

return number of occurrences of value

**Return Value**  
integer**extend**(*L*, *iterable*)

extend list by appending elements from the iterable

**index**(...)*L.index*(value, [start, [stop]]) -> integer – return first index of value**insert**(*L*, *index*, *object*)

insert object before index

**pack**(*self*, *val*, *fmt*=`DWORD`, *start*=0)Encode *val* and write it to buffer.**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex**(*cls*, *num*)Return the hex representation of *num* without the 0x prefix.If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.**pop**(*L*, *index*=...)

remove and return item at index (default last)

**Return Value**  
item**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)

reverse \*IN PLACE\*

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort \*IN PLACE\*; *cmp*(*x*, *y*) -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>

**start:** Position in buffer from which to decode

---

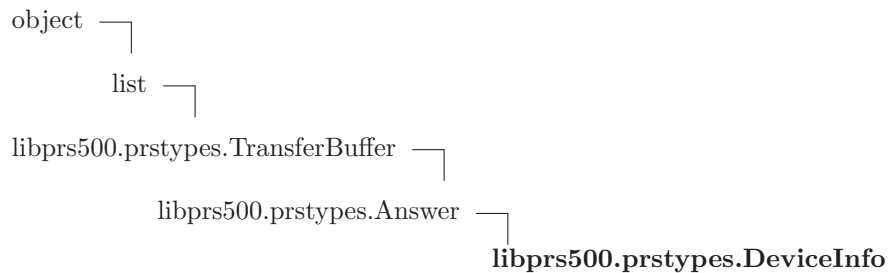
<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

#### 4.19.2 Properties

Name	Description
id	<b>Value:</b> <property object at 0x406aeb4>
__class__	<b>Value:</b> <attribute '__class__' of 'object' objects>

#### 4.20 Class DeviceInfo



Defines the structure of the packet containing information about the device

##### 4.20.1 Methods

---

**\_\_add\_\_**(*self*, *tb*)

---

Return a TransferBuffer rather than a list as the sum

Overrides: list.\_\_add\_\_

---



---

**\_\_contains\_\_**(*x*, *y*)

---

y in x

---



---

**\_\_delattr\_\_**(...)

---

x.\_\_delattr\_\_('name') <==> del x.name

---



---

**\_\_delitem\_\_**(*x*, *y*)

---

del x[y]

---

**`--delslice--`**(*x*, *i*, *j*)

`del x[i:j]`

Use of negative indices is not supported.

**`--eq--`**(*x*, *y*)

`x==y`

**`--ge--`**(*x*, *y*)

`x>=y`

**`--getattr--`**(...)

`x._getattr_('name') <==> x.name`

Overrides: `object._getattr--`

**`--getitem--`**(*x*, *y*)

`x[y]`

**`--getslice--`**(*self*, *start*, *end*)

Return a `TransferBuffer` rather than a list as the slice

Overrides: `list._getslice--`

**`--gt--`**(*x*, *y*)

`x>y`

**`--hash--`**(*x*)

`hash(x)`

Overrides: `object._hash--`

**`--iadd--`**(*x*, *y*)

`x+=y`

**`--imul--`**(*x*, *y*)

`x*=y`

**`--init--`**(*self*, *packet*)

#### Parameters

`packet`: `len(packet) ≥ 16`

Overrides: `libprs500.prstypes.TransferBuffer._init--`

`__iter__(x)``iter(x)``__le__(x, y)``x<=y``__len__(x)``len(x)``__lt__(x, y)``x<y``__mul__(x, n)``x*n``__ne__(x, y)``x!=y``__new__(T, S, ...)`**Return Value**a new object with type `S`, a subtype of `T`Overrides: `object.__new__``__reduce__()`

helper for pickle

`__reduce_ex__()`

helper for pickle

`__repr__(x)``repr(x)`Overrides: `object.__repr__``__reversed__(L)`

return a reverse iterator over the list

`__rmul__(x, n)``n*x`

**\_\_setattr\_\_**(...)

x.\_\_setattr\_\_('name', value) &lt;==&gt; x.name = value

**\_\_setitem\_\_**(x, i, y)

x[i]=y

**\_\_setslice\_\_**(x, i, j, y)

x[i:j]=y

Use of negative indices is not supported.

**\_\_str\_\_**(self)

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

0700 0100 0000 0000 0000 0000 0c00 0000	.....
0200 0000 0400 0000 4461 7461	.....Data

Overrides: object.\_\_str\_\_

**append**(L, object)

append object to end

**count**(L, value)

return number of occurrences of value

**Return Value**

integer

**extend**(L, iterable)

extend list by appending elements from the iterable

**index**(...)

L.index(value, [start, [stop]]) -&gt; integer – return first index of value

**insert**(L, index, object)

insert object before index

---

**pack**(*self*, *val*, *fmt*=DWORD, *start*=0)

---

Encode *val* and write it to buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer at which to write encoded data

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

---

**phex**(*cls*, *num*)

---

Return the hex representation of *num* without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in `TransferBuffer.__str__`.

---

**pop**(*L*, *index*=...)

---

remove and return item at index (default last)

**Return Value**

item

---

**remove**(*L*, *value*)

---

remove first occurrence of value

---

**reverse**(*L*)

---

reverse \*IN PLACE\*

---

**sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)

---

stable sort \*IN PLACE\*; *cmp*(*x*, *y*) -> -1, 0, 1

---

**unpack**(*self*, *fmt*=DWORD, *start*=0)

---

Return decoded data from buffer.

**Parameters**

**fmt:** See struct<sup>a</sup>  
**start:** Position in buffer from which to decode

---

<sup>a</sup><http://docs.python.org/lib/module-struct.html>

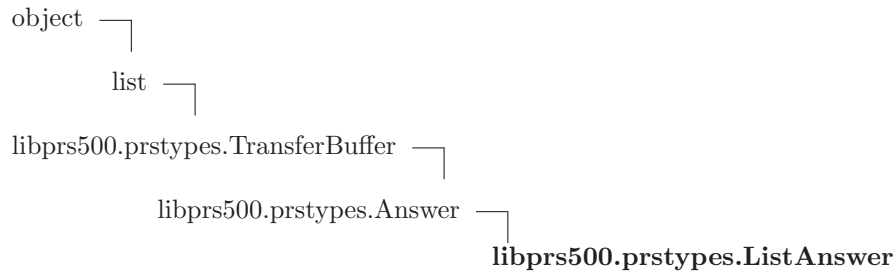
#### 4.20.2 Properties

Name	Description
<code>device_name</code>	<b>Value:</b> <property object at 0x406aec34>
<code>device_version</code>	<b>Value:</b> <property object at 0x406aec5c>
<code>software_version</code>	<b>Value:</b> <property object at 0x406aec84>
<code>mime_type</code>	<b>Value:</b> <property object at 0x406aecac>
<code>__class__</code>	<b>Value:</b> <attribute '__class__' of 'object' objects>

*continued on next page*

Name	Description
id	<b>Value:</b> <property object at 0x406aeaa4>

## 4.21 Class ListAnswer



Defines the structure of packets that contain items in a list.

### 4.21.1 Methods

**`__add__(self, tb)`**

Return a TransferBuffer rather than a list as the sum

Overrides: list.\_\_add\_\_

**`__contains__(x, y)`**

y in x

**`__delattr__(...)`**

x.\_\_delattr\_\_('name') <==> del x.name

**`__delitem__(x, y)`**

del x[y]

**`__delslice__(x, i, j)`**

del x[i:j]

Use of negative indices is not supported.

**`__eq__(x, y)`**

x==y

**`__ge__(x, y)`**

x>=y

**\_\_getattr\_\_**(...)`x.__getattr__('name') <==> x.name`Overrides: `object.__getattr__`**\_\_getitem\_\_**(*x*, *y*)`x[y]`**\_\_getslice\_\_**(*self*, *start*, *end*)

Return a TransferBuffer rather than a list as the slice

Overrides: `list.__getslice__`**\_\_gt\_\_**(*x*, *y*)`x > y`**\_\_hash\_\_**(*x*)`hash(x)`Overrides: `object.__hash__`**\_\_iadd\_\_**(*x*, *y*)`x += y`**\_\_imul\_\_**(*x*, *y*)`x *= y`**\_\_init\_\_**(*self*, *packet*)**Parameters**`packet: len(packet) ≥ 16`Overrides: `libprs500.prstypes.TransferBuffer.__init__`**\_\_iter\_\_**(*x*)`iter(x)`**\_\_le\_\_**(*x*, *y*)`x ≤ y`**\_\_len\_\_**(*x*)`len(x)`

**`__lt__`**(*x*, *y*)*x* < *y***`__mul__`**(*x*, *n*)*x* \* *n***`__ne__`**(*x*, *y*)*x* != *y***`__new__`**(*T*, *S*, ...)**Return Value**a new object with type *S*, a subtype of *T*Overrides: `object.__new__`**`__reduce__`**(...)

helper for pickle

**`__reduce_ex__`**(...)

helper for pickle

**`__repr__`**(*x*)`repr`(*x*)Overrides: `object.__repr__`**`__reversed__`**(*L*)

return a reverse iterator over the list

**`__rmul__`**(*x*, *n*)*n* \* *x***`__setattr__`**(...)*x*.`__setattr__`('name', value) <==> *x*.name = value**`__setitem__`**(*x*, *i*, *y*)*x*[*i*] = *y***`__setslice__`**(*x*, *i*, *j*, *y*)*x*[*i*:*j*] = *y*

Use of negative indices is not supported.

**\_\_str\_\_(self)**

Return a string representation of this buffer.

Packets are represented as hex strings, in 2-byte pairs,  $\leq 16$  bytes to a line. An ASCII representation is included. For example:

```

    0700 0100 0000 0000 0000 0000 0c00 0000      .....
    0200 0000 0400 0000 4461 7461                .....Data

```

Overrides: `object.__str__`**append(L, object)**

append object to end

**count(L, value)**

return number of occurrences of value

**Return Value****integer****extend(L, iterable)**

extend list by appending elements from the iterable

**index(...)**`L.index(value, [start, [stop]])` -> integer – return first index of value**insert(L, index, object)**

insert object before index

**pack(self, val, fmt=DWORD, start=0)**Encode `val` and write it to buffer.**Parameters****fmt:** See struct<sup>a</sup>**start:** Position in buffer at which to write encoded data<sup>a</sup><http://docs.python.org/lib/module-struct.html>**phex(cls, num)**Return the hex representation of `num` without the 0x prefix.

If the hex representation is only 1 digit it is padded to the left with a zero. Used in

`TransferBuffer.__str__`**pop(L, index=...)**

remove and return item at index (default last)

**Return Value****item**

**remove**(*L*, *value*)

remove first occurrence of value

**reverse**(*L*)reverse *\*IN PLACE\****sort**(*L*, *cmp*=None, *key*=None, *reverse*=False)stable sort *\*IN PLACE\**; *cmp*(*x*, *y*) -> -1, 0, 1**unpack**(*self*, *fmt*=DWORD, *start*=0)

Return decoded data from buffer.

**Parameters****fmt**: See struct<sup>a</sup>**start**: Position in buffer from which to decode<sup>a</sup><http://docs.python.org/lib/module-struct.html>**4.21.2 Properties**

Name	Description
<code>is_dir</code>	<b>Value:</b> <property object at 0x406aecfc>
<code>name_length</code>	<b>Value:</b> <property object at 0x406aed24>
<code>name</code>	<b>Value:</b> <property object at 0x406aed4c>
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>
<code>id</code>	<b>Value:</b> <property object at 0x406aeaa4>

## 5 Module `libprs500.terminfo`

### 5.1 Class `TerminalController`

A class that can be used to portably generate formatted output to a terminal.

‘`TerminalController`’ defines a set of instance variables whose values are initialized to the control sequence necessary to perform a given action. These can be simply included in normal output to the terminal:

```
>>> term = TerminalController()
>>> print 'This is '+term.GREEN+'green'+term.NORMAL
```

Alternatively, the ‘`render()`’ method can be used, which replaces ‘`${action}`’ with the string required to perform ‘`action`’:

```
>>> term = TerminalController()
>>> print term.render('This is ${GREEN}green${NORMAL}')
```

If the terminal doesn’t support a given action, then the value of the corresponding instance variable will be set to “”. As a result, the above code will still work on terminals that do not support color, except that their output will not be colored. Also, this means that you can test whether the terminal supports a given action by simply testing the truth value of the corresponding instance variable:

```
>>> term = TerminalController()
>>> if term.CLEAR_SCREEN:
...     print 'This terminal supports clearing the screen.'
```

Finally, if the width and height of the terminal are known, then they will be stored in the ‘`COLS`’ and ‘`LINES`’ attributes.

#### 5.1.1 Methods

<b><code>__init__(self, term_stream=sys.stdout)</code></b>
Create a ‘ <code>TerminalController</code> ’ and initialize its attributes with appropriate values for the current terminal. ‘ <code>term_stream</code> ’ is the stream that will be used for terminal output; if this stream is not a tty, then the terminal is assumed to be a dumb terminal (i.e., have no capabilities).

<b><code>render(self, template)</code></b>
Replace each \$-substitutions in the given template string with the corresponding terminal control string (if it’s defined) or “” (if it’s not).

#### 5.1.2 Class Variables

Name	Description
BOL	Move the cursor to the beginning of the line <b>Value:</b> ‘ ’
UP	Move the cursor up one line <b>Value:</b> ‘ ’
DOWN	Move the cursor down one line <b>Value:</b> ‘ ’

*continued on next page*

Name	Description
LEFT	Move the cursor left one char <b>Value:</b> ''
RIGHT	Move the cursor right one char <b>Value:</b> ''
CLEAR_SCREEN	Clear the screen and move to home position <b>Value:</b> ''
CLEAR_EOL	Clear to the end of the line. <b>Value:</b> ''
CLEAR_BOL	Clear to the beginning of the line. <b>Value:</b> ''
CLEAR_EOS	Clear to the end of the screen <b>Value:</b> ''
BOLD	Turn on bold mode <b>Value:</b> ''
BLINK	Turn on blink mode <b>Value:</b> ''
DIM	Turn on half-bright mode <b>Value:</b> ''
REVERSE	Turn on reverse-video mode <b>Value:</b> ''
NORMAL	Turn off all modes <b>Value:</b> ''
HIDE_CURSOR	Make the cursor invisible <b>Value:</b> ''
SHOW_CURSOR	Make the cursor visible <b>Value:</b> ''
COLS	Width of the terminal (None for unknown) <b>Value:</b> None
LINES	Height of the terminal (None for unknown) <b>Value:</b> None
WHITE	<b>Value:</b> ''
YELLOW	<b>Value:</b> ''
MAGENTA	<b>Value:</b> ''
RED	<b>Value:</b> ''
CYAN	<b>Value:</b> ''
GREEN	<b>Value:</b> ''
BLUE	<b>Value:</b> ''
BLACK	<b>Value:</b> ''
BG_CYAN	<b>Value:</b> ''
BG_GREEN	<b>Value:</b> ''
BG_BLUE	<b>Value:</b> ''
BG_BLACK	<b>Value:</b> ''
BG_WHITE	<b>Value:</b> ''
BG_YELLOW	<b>Value:</b> ''
BG_MAGENTA	<b>Value:</b> ''
BG_RED	<b>Value:</b> ''

## 5.2 Class `ProgressBar`

A 3-line progress bar, which looks like:

```

                                Header
20% [=====-----]
                                progress message

```

The progress bar is colored, if the terminal supports color output; and adjusts to the width of the terminal.

### 5.2.1 Methods

```
__init__(self, term, header)
```

```
update(self, percent, message)
```

```
clear(self)
```

### 5.2.2 Class Variables

Name	Description
BAR	<b>Value:</b> <code>'%3d%% \${GREEN} [\${BOLD}%s\${NORMAL}\${GREEN}] \${NORMAL}\n'</code>
HEADER	<b>Value:</b> <code>'\${BOLD}\${CYAN}%s\${NORMAL}\n\n'</code>

### 5.2.3 Instance Variables

Name	Description
cleared	true if we haven't drawn the bar yet.

## 6 Module prs500

Provides a command-line interface to the SONY Reader PRS-500.

For usage information run the script.

### 6.1 Functions

<code>info(<i>dev</i>)</code>
-------------------------------

<code>ls(<i>dev</i>, <i>path</i>, <i>term</i>, <i>recurse</i>=False, <i>color</i>=False, <i>human_readable_size</i>=False, <i>ll</i>=False, <i>cols</i>=0)</code>
---

<code>main()</code>
---------------------

### 6.2 Variables

Name	Description
MINIMUM_COL_WIDTH	Minimum width of columns in ls output <b>Value:</b> 12

### 6.3 Class FileFormatter

```

object └─
          prs500.FileFormatter

```

#### 6.3.1 Methods

<code>__init__(<i>self</i>, <i>file</i>, <i>term</i>)</code> <code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)
---

<code>__delattr__(...)</code> <code>x.__delattr__('name')</code> <==> <code>del x.name</code>
--

<code>__getattr__(...)</code> <code>x.__getattr__('name')</code> <==> <code>x.name</code>
--

<code>__hash__(<i>x</i>)</code> <code>hash(x)</code>
---

**`--new--`**(*T*, *S*, ...)

**Return Value**

a new object with type *S*, a subtype of *T*

**`--reduce--`**(...)

helper for pickle

**`--reduce_ex--`**(...)

helper for pickle

**`--repr--`**(*x*)

`repr(x)`

**`--setattr--`**(...)

`x.__setattr__('name', value) <==> x.name = value`

**`--str--`**(*x*)

`str(x)`

### 6.3.2 Properties

Name	Description
<code>mode_string</code>	<b>Value:</b> <property object at 0x406f9374>
<code>name_in_color</code>	<b>Value:</b> <property object at 0x406f939c>
<code>human_readable_size</code>	<b>Value:</b> <property object at 0x406f93c4>
<code>modification_time</code>	<b>Value:</b> <property object at 0x406f93ec>
<code>creation_time</code>	<b>Value:</b> <property object at 0x406f9414>
<code>--class--</code>	<b>Value:</b> <attribute ' <code>--class--</code> ' of 'object' objects>

## 7 Module struct

Functions to convert between Python values and C structs.  
Python strings are used to hold the data representing the C struct  
and also as format strings to describe the layout of data in the C struct.

The optional first format char indicates byte order, size and alignment:

- @: native order, size & alignment (default)
- =: native order, std. size & alignment
- <: little-endian, std. size & alignment
- >: big-endian, std. size & alignment
- !: same as >

The remaining chars indicate types of args and must match exactly;  
these can be preceded by a decimal repeat count:

- x: pad byte (no data); c:char; b:signed byte; B:unsigned byte;
- h:short; H:unsigned short; i:int; I:unsigned int;
- l:long; L:unsigned long; f:float; d:double.

Special cases (preceding decimal count indicates length):

- s:string (array of char); p: pascal string (with count byte).

Special case (only available in native format):

- P:an integer type that is wide enough to hold a pointer.

Special case (not in native mode unless 'long long' in platform C):

- q:long long; Q:unsigned long long

Whitespace between formats is ignored.

The variable struct.error is an exception raised on errors.

### 7.1 Functions

<b>calcsize(<i>fmt</i>)</b>
Return size of C struct described by format string <i>fmt</i> . See struct.__doc__ for more on format strings.

<b>pack(<i>fmt</i>, *<i>args</i>)</b>
Return string containing values <i>v1</i> , <i>v2</i> , ... packed according to <i>fmt</i> . See struct.__doc__ for more on format strings.

<b>pack_into(<i>fmt</i>, <i>buf</i>, <i>offset</i>, *<i>args</i>)</b>
Pack the values <i>v1</i> , <i>v2</i> , ... according to <i>fmt</i> , write the packed bytes into the writable buffer <i>buf</i> starting at <i>offset</i> . See struct.__doc__ for more on format strings.

<b>unpack(<i>fmt</i>, <i>s</i>)</b>
Unpack the string, containing packed C structure data, according to <i>fmt</i> . Requires len(string)==calcsize( <i>fmt</i> ). See struct.__doc__ for more on format strings.

**unpack\_from**(*fmt*, *buf*, *offset*=0)

Unpack the buffer, containing packed C structure data, according to *fmt* starting at *offset*. Requires `len(buffer[offset:]) >= calcsize(fmt)`. See `struct.__doc__` for more on format strings.

## 7.2 Variables

Name	Description
<code>__version__</code>	<b>Value:</b> <code>'0.1'</code>

## 8 Module *usb*

USB access module

### 8.1 Functions

<b>busses(...)</b>
Returns a tuple with the usb busses

### 8.2 Variables

Name	Description
CLASS_AUDIO	Value: 1
CLASS_COMM	Value: 2
CLASS_DATA	Value: 10
CLASS_HID	Value: 3
CLASS_HUB	Value: 9
CLASS_MASS_STORAGE	Value: 8
CLASS_PER_INTERFACE	Value: 0
CLASS_PRINTER	Value: 7
CLASS_VENDOR_SPEC	Value: 255
DT_CONFIG	Value: 2
DT_CONFIG_SIZE	Value: 9
DT_DEVICE	Value: 1
DT_DEVICE_SIZE	Value: 18
DT_ENDPOINT	Value: 5
DT_ENDPOINT_AUDIO_SIZE	Value: 9
DT_ENDPOINT_SIZE	Value: 7
DT_HID	Value: 33
DT_HUB	Value: 41
DT_HUB_NONVAR_SIZE	Value: 7
DT_INTERFACE	Value: 4
DT_INTERFACE_SIZE	Value: 9
DT_PHYSICAL	Value: 35
DT_REPORT	Value: 34
DT_STRING	Value: 3
ENDPOINT_ADDRESS_MASK	Value: 15
ENDPOINT_DIR_MASK	Value: 128
ENDPOINT_IN	Value: 128
ENDPOINT_OUT	Value: 0
ENDPOINT_TYPE_BULK	Value: 2
ENDPOINT_TYPE_CONTROL	Value: 0
ENDPOINT_TYPE_INTERRUPT	Value: 3
ENDPOINT_TYPE_ISOCHRONOUS	Value: 1

*continued on next page*

Name	Description
ENDPOINT_TYPE_MASK	Value: 3
ERROR_BEGIN	Value: 500000
MAXALTSETTING	Value: 128
MAXCONFIG	Value: 8
MAXENDPOINTS	Value: 32
MAXINTERFACES	Value: 32
RECIP_DEVICE	Value: 0
RECIP_ENDPOINT	Value: 2
RECIP_INTERFACE	Value: 1
RECIP_OTHER	Value: 3
REQ_CLEAR_FEATURE	Value: 1
REQ_GET_CONFIGURATIO- N	Value: 8
REQ_GET_DESCRIPTOR	Value: 6
REQ_GET_INTERFACE	Value: 10
REQ_GET_STATUS	Value: 0
REQ_SET_ADDRESS	Value: 5
REQ_SET_CONFIGURATIO- N	Value: 9
REQ_SET_DESCRIPTOR	Value: 7
REQ_SET_FEATURE	Value: 3
REQ_SET_INTERFACE	Value: 11
REQ_SYNCH_FRAME	Value: 12
TYPE_CLASS	Value: 32
TYPE_RESERVED	Value: 96
TYPE_STANDARD	Value: 0
TYPE_VENDOR	Value: 64

### 8.3 Class Bus

```

object └─
         usb.Bus

```

Bus object

#### 8.3.1 Methods

```

__delattr__(...)
x.__delattr__('name') <==> del x.name

```

```

__getattr__(...)
x.__getattr__('name') <==> x.name

```

```

__hash__(x)
hash(x)

```

<b><code>__init__</code></b> (...)
<code>x.__init__</code> (...) initializes x; see <code>x.__class__.__doc__</code> for signature

<b><code>__new__</code></b> ( <i>T</i> , <i>S</i> , ...)
<b>Return Value</b> a new object with type <i>S</i> , a subtype of <i>T</i>
Overrides: <code>object.__new__</code>

<b><code>__reduce__</code></b> (...)
helper for pickle

<b><code>__reduce_ex__</code></b> (...)
helper for pickle

<b><code>__repr__</code></b> ( <i>x</i> )
<code>repr</code> ( <i>x</i> )

<b><code>__setattr__</code></b> (...)
<code>x.__setattr__</code> ('name', value) <==> <code>x.name</code> = value

<b><code>__str__</code></b> ( <i>x</i> )
<code>str</code> ( <i>x</i> )

### 8.3.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>

### 8.3.3 Class Variables

Name	Description
<code>devices</code>	<b>Value:</b> <member ' <code>devices</code> ' of ' <code>usb.Bus</code> ' objects>
<code>dirname</code>	<b>Value:</b> <member ' <code>dirname</code> ' of ' <code>usb.Bus</code> ' objects>
<code>location</code>	<b>Value:</b> <member ' <code>location</code> ' of ' <code>usb.Bus</code> ' objects>

## 8.4 Class Configuration

```

object └─
          usb.Configuration
Configuration descriptor object

```

### 8.4.1 Methods

<code>__delattr__(...)</code>
<code>x.__delattr__('name') &lt;==&gt; del x.name</code>

<code>__getattr__(...)</code>
<code>x.__getattr__('name') &lt;==&gt; x.name</code>

<code>__hash__(x)</code>
<code>hash(x)</code>

<code>__init__(...)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature

<code>__new__(T, S, ...)</code>
<b>Return Value</b>
a new object with type <code>S</code> , a subtype of <code>T</code>
Overrides: <code>object.__new__</code>

<code>__reduce__(...)</code>
helper for pickle

<code>__reduce_ex__(...)</code>
helper for pickle

<code>__repr__(x)</code>
<code>repr(x)</code>

<code>__setattr__(...)</code>
<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code>

<code>__str__(x)</code>
<code>str(x)</code>

### 8.4.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute <code>'__class__'</code> of <code>'object'</code> objects>

### 8.4.3 Class Variables

Name	Description
<code>iConfiguration</code>	<b>Value:</b> <member 'iConfiguration' of 'usb.Configuration' objects>
<code>interfaces</code>	<b>Value:</b> <member 'interfaces' of 'usb.Configuration' objects>
<code>maxPower</code>	<b>Value:</b> <member 'maxPower' of 'usb.Configuration' objects>
<code>remoteWakeup</code>	<b>Value:</b> <member 'remoteWakeup' of 'usb.Configuration' objects>
<code>selfPowered</code>	<b>Value:</b> <member 'selfPowered' of 'usb.Configuration' objects>
<code>totalLength</code>	<b>Value:</b> <member 'totalLength' of 'usb.Configuration' objects>
<code>value</code>	<b>Value:</b> <member 'value' of 'usb.Configuration' objects>

## 8.5 Class Device

```

object └─
         └─ usb.Device

```

Device descriptor object

### 8.5.1 Methods

<b><code>__delattr__(...)</code></b>
<code>x.__delattr__('name')</code> <==> <code>del x.name</code>
<b><code>__getattr__(...)</code></b>
<code>x.__getattr__('name')</code> <==> <code>x.name</code>
<b><code>__hash__(x)</code></b>
<code>hash(x)</code>
<b><code>__init__(...)</code></b>
<code>x.__init__(...)</code> initializes x; see <code>x.__class__.__doc__</code> for signature
<b><code>__new__(T, S, ...)</code></b>
<b>Return Value</b>
a new object with type S, a subtype of T
Overrides: <code>object.__new__</code>

**\_\_reduce\_\_**(...)

helper for pickle

**\_\_reduce\_ex\_\_**(...)

helper for pickle

**\_\_repr\_\_**(*x*)repr(*x*)**\_\_setattr\_\_**(...)*x*.\_\_setattr\_\_('name', value) <==> *x*.name = value**\_\_str\_\_**(*x*)str(*x*)**open**()

Open the device for use. Returns a DeviceHandle object.

**Return Value**

DeviceHandle

### 8.5.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of ' <code>object</code> ' objects>

### 8.5.3 Class Variables

Name	Description
<code>configurations</code>	<b>Value:</b> <member ' <code>configurations</code> ' of ' <code>usb.Device</code> ' objects>
<code>deviceClass</code>	<b>Value:</b> <member ' <code>deviceClass</code> ' of ' <code>usb.Device</code> ' objects>
<code>deviceProtocol</code>	<b>Value:</b> <member ' <code>deviceProtocol</code> ' of ' <code>usb.Device</code> ' objects>
<code>deviceSubClass</code>	<b>Value:</b> <member ' <code>deviceSubClass</code> ' of ' <code>usb.Device</code> ' objects>
<code>deviceVersion</code>	<b>Value:</b> <member ' <code>deviceVersion</code> ' of ' <code>usb.Device</code> ' objects>
<code>filename</code>	<b>Value:</b> <member ' <code>filename</code> ' of ' <code>usb.Device</code> ' objects>
<code>iManufacturer</code>	<b>Value:</b> <member ' <code>iManufacturer</code> ' of ' <code>usb.Device</code> ' objects>
<code>iProduct</code>	<b>Value:</b> <member ' <code>iProduct</code> ' of ' <code>usb.Device</code> ' objects>
<code>iSerialNumber</code>	<b>Value:</b> <member ' <code>iSerialNumber</code> ' of ' <code>usb.Device</code> ' objects>

*continued on next page*

Name	Description
<code>idProduct</code>	<b>Value:</b> <member 'idProduct' of 'usb.Device' objects>
<code>idVendor</code>	<b>Value:</b> <member 'idVendor' of 'usb.Device' objects>
<code>maxPacketSize</code>	<b>Value:</b> <member 'maxPacketSize' of 'usb.Device' objects->
<code>usbVersion</code>	<b>Value:</b> <member 'usbVersion' of 'usb.Device' objects>

## 8.6 Class *DeviceHandle*



DeviceHandle object

### 8.6.1 Methods

<b><code>__delattr__</code></b> (...)
<code>x.__delattr__('name')</code> <==> <code>del x.name</code>
<b><code>__getattr__</code></b> (...)
<code>x.__getattr__('name')</code> <==> <code>x.name</code>
<b><code>__hash__</code></b> ( <i>x</i> )
<code>hash(x)</code>
<b><code>__init__</code></b> (...)
<code>x.__init__()</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
<b><code>__new__</code></b> ( <i>T</i> , <i>S</i> , ...)
<b>Return Value</b> a new object with type <i>S</i> , a subtype of <i>T</i> Overrides: <code>object.__new__</code>
<b><code>__reduce__</code></b> (...)
helper for pickle
<b><code>__reduce_ex__</code></b> (...)
helper for pickle
<b><code>__repr__</code></b> ( <i>x</i> )
<code>repr(x)</code>

---

**\_\_setattr\_\_**(...)

---

 x.\_\_setattr\_\_('name', value) <==> x.name = value

---

**\_\_str\_\_**(x)

---

 str(x)

---

**bulkRead**(*endpoint*, *size*, *timeout*=100)

---

 Performs a bulk read request to the endpoint specified.

Arguments:

*endpoint*: endpoint number.  
     *size*: number of bytes to read.  
     *timeout*: operation timeout in milliseconds. (default: 100)

Returns a tuple with the data read.

**Return Value**

buffer

---

**bulkWrite**(*endpoint*, *buffer*, *timeout*=100)

---

 Performs a bulk write request to the endpoint specified.

Arguments:

*endpoint*: endpoint number.  
     *buffer*: sequence data buffer to write.  
         This parameter can be any sequence type  
     *timeout*: operation timeout in milliseconds. (default: 100)

Returns the number of bytes written.

**Return Value**

bytesWritten

---

**claimInterface**(*interface*)

---

 Claims the interface with the Operating System.

Arguments:

*interface*: interface number or an Interface object.

**Return Value**

None

---

**clearHalt**(*endpoint*)

---

 Clears any halt status on the specified endpoint.

Arguments:

*endpoint*: endpoint number.

**Return Value**

None

---

**controlMsg**(*requestType*, *request*, *buffer*, *value*=0, *index*=0, *timeout*=100)

---

Performs a control request to the default control pipe on a device.

Arguments:

*requestType*: specifies the direction of data flow, the type of request, and the recipient.  
    *request*: specifies the request.  
    *buffer*: if the transfer is a write transfer, *buffer* is a sequence with the transfer data, otherwise, *buffer* is the number of bytes to read.  
    *value*: specific information to pass to the device. (default: 0)  
    *index*: specific information to pass to the device. (default: 0)  
    *timeout*: operation timeout in miliseconds. (default: 100)

Returns the number of bytes written.

**Return Value**

bytesWritten|buffer

---

**getDescriptor**(...)

---

getDescriptor(*type*, *index*, *len*, *endpoint* = -1) -> descriptor

Retrieves a descriptor from the device identified by the *type* and *index* of the descriptor.

Arguments:

*type*: descriptor type.  
    *index*: index of the descriptor.  
    *len*: descriptor length.  
    *endpoint*: endpoint number from descriptor is read. If it is omitted, the descriptor is read from default control pipe.

---

**getString**(...)

---

getString(*index*, *len*, *langid* = -1) -> string

Retrieves the string descriptor specified by *index* and *langid* from a device.

Arguments:

*index*: index of descriptor in the device.  
    *len*: number of bytes of the string  
    *langid*: Language ID. If it is omitted, will be used the first language.

**interruptRead**(*endpoint, size, timeout=100*)

Performs a interrupt read request to the endpoint specified.

Arguments:

*endpoint*: endpoint number.  
    *size*: number of bytes to read.  
    *timeout*: operation timeout in miliseconds. (default: 100)

Returns a tuple with the data read.

**Return Value**

*buffer*

**interruptWrite**(*endpoint, buffer, timeout=100*)

Performs a interrupt write request to the endpoint specified.

Arguments:

*endpoint*: endpoint number.  
    *buffer*: sequence data buffer to write.  
        This parameter can be any sequence type  
    *timeout*: operation timeout in miliseconds. (default: 100)

Returns the number of bytes written.

**Return Value**

*bytesWritten*

**releaseInterface**()

Releases an interface previously claimed with *claimInterface*.

**Return Value**

    None

**reset**()

Resets the specified device by sending a RESET down the port it is connected to.

**Return Value**

    None

**resetEndpoint**(*endpoint*)

Resets all state (like toggles) for the specified endpoint.

Arguments:

*endpoint*: endpoint number.

**Return Value**

    None

**setAltInterface**(*alternate*)

Sets the active alternate setting of the current interface.

Arguments:

*alternate*: an alternate setting number or an Interface object.

**Return Value**

    None

<b>setConfiguration</b> ( <i>configuration</i> )
Sets the active configuration of a device.
Arguments:
<i>configuration</i> : a configuration value or a Configuration object.
<b>Return Value</b>
None

### 8.6.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute ' <code>__class__</code> ' of 'object' objects>

## 8.7 Class Endpoint



Endpoint descriptor object

### 8.7.1 Methods

<b>__delattr__</b> (...)
<code>x.__delattr__('name')</code> <==> <code>del x.name</code>
<b>__getattr__</b> (...)
<code>x.__getattr__('name')</code> <==> <code>x.name</code>
<b>__hash__</b> ( <i>x</i> )
<code>hash(x)</code>
<b>__init__</b> (...)
<code>x.__init__()</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
<b>__new__</b> ( <i>T, S, ...</i> )
<b>Return Value</b>
a new object with type <code>S</code> , a subtype of <code>T</code>
Overrides: <code>object.__new__</code>
<b>__reduce__</b> (...)
helper for pickle

<code>__reduce_ex__(...)</code>
---------------------------------

helper for pickle
-------------------

<code>__repr__(x)</code>
--------------------------

<code>repr(x)</code>
----------------------

<code>__setattr__(...)</code>
-------------------------------

<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code>
---

<code>__str__(x)</code>
-------------------------

<code>str(x)</code>
---------------------

### 8.7.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute <code>'__class__'</code> of <code>'object'</code> objects>

### 8.7.3 Class Variables

Name	Description
<code>address</code>	<b>Value:</b> <member <code>'address'</code> of <code>'usb.Endpoint'</code> objects>
<code>interval</code>	<b>Value:</b> <member <code>'interval'</code> of <code>'usb.Endpoint'</code> objects>
<code>maxPacketSize</code>	<b>Value:</b> <member <code>'maxPacketSize'</code> of <code>'usb.Endpoint'</code> objects>
<code>type</code>	<b>Value:</b> <member <code>'type'</code> of <code>'usb.Endpoint'</code> objects>

## 8.8 Class Interface

```

object ┌
      │ usb.Interface

```

Interface descriptor object

### 8.8.1 Methods

<code>__delattr__(...)</code>
-------------------------------

<code>x.__delattr__('name') &lt;==&gt; del x.name</code>
--

<code>__getattr__(...)</code>
-------------------------------

<code>x.__getattr__('name') &lt;==&gt; x.name</code>
--

<code>__hash__(x)</code>
<code>hash(x)</code>
<code>__init__(...)</code>
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
<code>__new__(T, S, ...)</code>
<b>Return Value</b> a new object with type <code>S</code> , a subtype of <code>T</code> Overrides: <code>object.__new__</code>
<code>__reduce__(...)</code>
helper for pickle
<code>__reduce_ex__(...)</code>
helper for pickle
<code>__repr__(x)</code>
<code>repr(x)</code>
<code>__setattr__(...)</code>
<code>x.__setattr__('name', value) &lt;==&gt; x.name = value</code>
<code>__str__(x)</code>
<code>str(x)</code>

### 8.8.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute <code>'__class__'</code> of <code>'object'</code> objects>

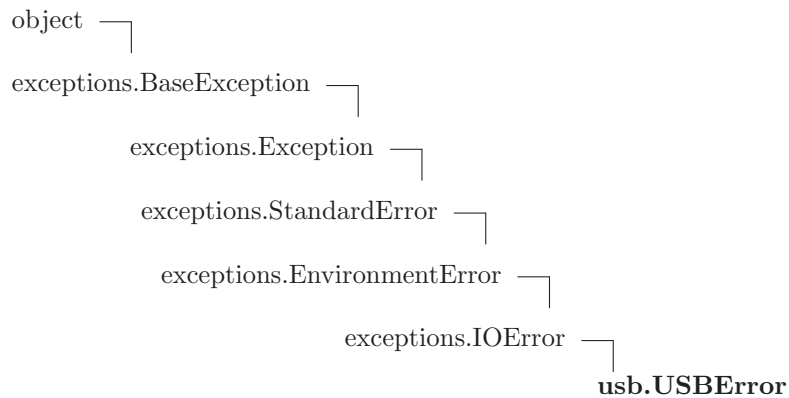
### 8.8.3 Class Variables

Name	Description
<code>alternateSetting</code>	<b>Value:</b> <member <code>'alternateSetting'</code> of <code>'usb.Interface'</code> objects>
<code>endpoints</code>	<b>Value:</b> <member <code>'endpoints'</code> of <code>'usb.Interface'</code> objects>
<code>interfaceClass</code>	<b>Value:</b> <member <code>'interfaceClass'</code> of <code>'usb.Interface'</code> objects>

*continued on next page*

Name	Description
<code>interfaceNumber</code>	<b>Value:</b> <member 'interfaceNumber' of 'usb.Interface' objects>
<code>interfaceProtocol</code>	<b>Value:</b> <member 'interfaceProtocol' of 'usb.Interface' - objects>
<code>interfaceSubClass</code>	<b>Value:</b> <member 'interfaceSubClass' of 'usb.Interface' - objects>

## 8.9 Class `USBError`



### 8.9.1 Methods

<b><code>__delattr__</code></b> (...)
<code>x.__delattr__('name')</code> <==> <code>del x.name</code>
Overrides: <code>object.__delattr__</code>

<b><code>__getattr__</code></b> (...)
<code>x.__getattr__('name')</code> <==> <code>x.name</code>
Overrides: <code>object.__getattr__</code>

<b><code>__getitem__</code></b> ( <i>x</i> , <i>y</i> )
<code>x[y]</code>

<b><code>__hash__</code></b> ( <i>x</i> )
<code>hash(x)</code>

<b><code>__init__</code></b> (...)
<code>x.__init__(...)</code> initializes <code>x</code> ; see <code>x.__class__.__doc__</code> for signature
Overrides: <code>exceptions.EnvironmentError.__init__</code>

**`__new__`**(*T*, *S*, ...)

**Return Value**

a new object with type *S*, a subtype of *T*

Overrides: `exceptions.EnvironmentError.__new__`

**`__reduce__`**(...)

Overrides: `exceptions.BaseException.__reduce__`

**`__reduce_ex__`**(...)

helper for pickle

**`__repr__`**(*x*)

`repr(x)`

Overrides: `object.__repr__`

**`__setattr__`**(...)

`x.__setattr__('name', value) <==> x.name = value`

Overrides: `object.__setattr__`

**`__setstate__`**(...)

**`__str__`**(*x*)

`str(x)`

Overrides: `exceptions.BaseException.__str__`

### 8.9.2 Properties

Name	Description
<code>__class__</code>	<b>Value:</b> <attribute <code>'__class__'</code> of <code>'object'</code> objects>
<code>args</code>	<b>Value:</b> <attribute <code>'args'</code> of <code>'exceptions.BaseException'</code> objects>

### 8.9.3 Class Variables

Name	Description
<code>errno</code>	<b>Value:</b> <member <code>'errno'</code> of <code>'exceptions.EnvironmentError'</code> objects>
<code>filename</code>	<b>Value:</b> <member <code>'filename'</code> of <code>'exceptions.EnvironmentError'</code> objects>
<code>message</code>	<b>Value:</b> <member <code>'message'</code> of <code>'exceptions.EnvironmentError'</code> objects>
<code>strerror</code>	<b>Value:</b> <member <code>'strerror'</code> of <code>'exceptions.EnvironmentError'</code> objects>

## Index

- exceptions.BaseException.\_\_getitem\_\_ (function), 11, 13, 15, 16, 18, 137
- exceptions.BaseException.\_\_setstate\_\_ (function), 12, 14, 15, 17, 19, 138
- libprs500 (package), 5
  - libprs500.communicate (module), 6–10
    - libprs500.communicate.DeviceDescriptor (class), 7
    - libprs500.communicate.File (class), 6–7
    - libprs500.communicate.PRS500Device (class), 7–10
  - libprs500.errors (module), 11–19
    - libprs500.errors.ArgumentError (class), 14–16
    - libprs500.errors.ControlError (class), 18–19
    - libprs500.errors.PacketError (class), 13–14
    - libprs500.errors.PathError (class), 16–18
    - libprs500.errors.ProtocolError (class), 11–13
  - libprs500.prstypes (module), 20–116
    - libprs500.prstypes.AcknowledgeBulkRead (class), 54–59
    - libprs500.prstypes.Answer (class), 93–98
    - libprs500.prstypes.Command (class), 25–30
    - libprs500.prstypes.DeviceInfo (class), 107–112
    - libprs500.prstypes.DeviceInfoQuery (class), 59–64
    - libprs500.prstypes.DirClose (class), 44–49
    - libprs500.prstypes.DirOpen (class), 34–39
    - libprs500.prstypes.DirRead (class), 39–44
    - libprs500.prstypes.FileClose (class), 64–68
    - libprs500.prstypes.FileOpen (class), 68–73
    - libprs500.prstypes.FileProperties (class), 98–102
    - libprs500.prstypes.FileRead (class), 73–78
    - libprs500.prstypes.IdAnswer (class), 102–107
    - libprs500.prstypes.ListAnswer (class), 112–116
    - libprs500.prstypes.ListResponse (class), 88–93
    - libprs500.prstypes.LongCommand (class), 49–54
    - libprs500.prstypes.PathQuery (class), 78–83
    - libprs500.prstypes.Response (class), 83–88
    - libprs500.prstypes.ShortCommand (class), 30–34
    - libprs500.prstypes.TransferBuffer (class), 20–25
  - libprs500.terminfo (module), 117–119
    - libprs500.terminfo.ProgressBar (class), 118–119
    - libprs500.terminfo.TerminalController (class), 117–118
  - list.\_\_contains\_\_ (function), 22, 25, 30, 35, 40, 45, 50, 55, 59, 64, 69, 74, 79, 84, 88, 94, 98, 103, 107, 112
  - list.\_\_delitem\_\_ (function), 22, 26, 30, 35, 40, 45, 50, 55, 59, 64, 69, 74, 79, 84, 89, 94, 98, 103, 107, 112
  - list.\_\_delslice\_\_ (function), 22, 26, 30, 35, 40, 45, 50, 55, 59, 64, 69, 74, 79, 84, 89, 94, 98, 103, 107, 112
  - list.\_\_eq\_\_ (function), 22, 26, 30, 35, 40, 45, 50, 55, 60, 64, 69, 74, 79, 84, 89, 94, 98, 103, 108, 112
  - list.\_\_ge\_\_ (function), 22, 26, 30, 35, 40, 45, 50, 55, 60, 64, 69, 74, 79, 84, 89, 94, 98, 103, 108, 112
  - list.\_\_getitem\_\_ (function), 22, 26, 31, 35, 40, 45, 50, 55, 60, 65, 69, 74, 79, 84, 89, 94, 99, 103, 108, 113
  - list.\_\_gt\_\_ (function), 22, 26, 31, 35, 41, 46, 51, 55, 60, 65, 69, 75, 79, 84, 89, 94, 99, 103, 108, 113
  - list.\_\_iadd\_\_ (function), 22, 26, 31, 36, 41, 46, 51, 56, 60, 65, 70, 75, 80, 85, 89, 95, 99, 104, 108, 113
  - list.\_\_imul\_\_ (function), 23, 26, 31, 36, 41, 46, 51, 56, 60, 65, 70, 75, 80, 85, 89, 95, 99, 104, 108, 113
  - list.\_\_iter\_\_ (function), 23, 26, 31, 36, 41, 46, 51, 56, 60, 65, 70, 75, 80, 85, 90, 95, 99, 104, 108, 113
  - list.\_\_le\_\_ (function), 23, 27, 31, 36, 41, 46, 51, 56, 60, 65, 70, 75, 80, 85, 90, 95, 99, 104, 109, 113
  - list.\_\_len\_\_ (function), 23, 27, 31, 36, 41, 46, 51, 56, 60, 65, 70, 75, 80, 85, 90, 95, 99, 104, 109, 113
  - list.\_\_lt\_\_ (function), 23, 27, 31, 36, 41, 46, 51, 56, 61, 65, 70, 75, 80, 85, 90, 95, 99, 104, 109, 113
  - list.\_\_mul\_\_ (function), 23, 27, 31, 36, 41, 46, 51, 56, 61, 65, 70, 75, 80, 85, 90, 95, 100, 104, 109, 114
  - list.\_\_ne\_\_ (function), 23, 27, 32, 36, 41, 46, 51, 56, 61, 65, 70, 75, 80, 85, 90, 95, 100, 104, 109, 114
  - list.\_\_reversed\_\_ (function), 23, 27, 32, 37, 42, 47, 52, 57, 61, 66, 71, 76, 81, 86, 90, 96, 100, 105, 109, 114
  - list.\_\_rmul\_\_ (function), 24, 27, 32, 37, 42, 47, 52, 57, 61, 66, 71, 76, 81, 86, 91, 96, 100, 105, 109, 114
  - list.\_\_setitem\_\_ (function), 24, 28, 32, 37, 42, 47, 52, 57, 61, 66, 71, 76, 81, 86, 91, 96, 100, 105, 110, 114

- `list.__setslice__` (*function*), 24, 28, 32, 37, 42, 47, 52, 57, 61, 66, 71, 76, 81, 86, 91, 96, 100, 105, 110, 114
- `list.append` (*function*), 24, 28, 33, 37, 42, 47, 52, 57, 62, 66, 71, 76, 81, 86, 91, 96, 101, 105, 110, 115
- `list.count` (*function*), 24, 28, 33, 37, 42, 47, 52, 57, 62, 67, 71, 76, 81, 86, 91, 96, 101, 105, 110, 115
- `list.extend` (*function*), 24, 28, 33, 37, 43, 48, 53, 57, 62, 67, 71, 77, 81, 86, 91, 96, 101, 106, 110, 115
- `list.index` (*function*), 24, 28, 33, 38, 43, 48, 53, 57, 62, 67, 72, 77, 82, 87, 91, 96, 101, 106, 110, 115
- `list.insert` (*function*), 24, 28, 33, 38, 43, 48, 53, 58, 62, 67, 72, 77, 82, 87, 91, 97, 101, 106, 110, 115
- `list.pop` (*function*), 24, 29, 33, 38, 43, 48, 53, 58, 62, 67, 72, 77, 82, 87, 92, 97, 101, 106, 111, 115
- `list.remove` (*function*), 24, 29, 33, 38, 43, 48, 53, 58, 63, 67, 72, 77, 82, 87, 92, 97, 101, 106, 111, 115
- `list.reverse` (*function*), 25, 29, 34, 38, 43, 48, 53, 58, 63, 67, 72, 77, 82, 87, 92, 97, 102, 106, 111, 116
- `list.sort` (*function*), 25, 29, 34, 38, 43, 48, 53, 58, 63, 67, 72, 77, 82, 87, 92, 97, 102, 106, 111, 116
- `object.__delattr__` (*function*), 6, 9, 22, 25, 30, 35, 40, 45, 50, 55, 59, 64, 69, 74, 79, 84, 89, 94, 98, 103, 107, 112, 120, 125, 127, 128, 130, 134, 135
- `object.__getattr__` (*function*), 6, 9, 120, 125, 127, 128, 130, 134, 135
- `object.__hash__` (*function*), 6, 9, 11, 13, 15, 17, 18, 120, 125, 127, 128, 130, 134, 135, 137
- `object.__init__` (*function*), 125, 127, 128, 130, 134, 136
- `object.__new__` (*function*), 6, 9, 120
- `object.__reduce__` (*function*), 6, 9, 23, 27, 32, 36, 41, 46, 51, 56, 61, 66, 70, 76, 80, 85, 90, 95, 100, 104, 109, 114, 121, 126–128, 130, 134, 136
- `object.__reduce_ex__` (*function*), 7, 9, 12, 13, 15, 17, 19, 23, 27, 32, 36, 42, 47, 52, 56, 61, 66, 70, 76, 80, 85, 90, 95, 100, 105, 109, 114, 121, 126, 127, 129, 130, 134, 136, 138
- `object.__repr__` (*function*), 9, 121, 126, 127, 129, 130, 135, 136
- `object.__setattr__` (*function*), 7, 9, 24, 27, 32, 37, 42, 47, 52, 57, 61, 66, 71, 76, 81, 86, 91, 96, 100, 105, 109, 114, 121, 126, 127, 129, 130, 135, 136
- `object.__str__` (*function*), 7, 9, 121, 126, 127, 129, 131, 135, 136
- `prs500` (*module*), 120–121
  - `prs500.FileFormatter` (*class*), 120–121
  - `prs500.info` (*function*), 120
  - `prs500.ls` (*function*), 120
  - `prs500.main` (*function*), 120
- `struct` (*module*), 122–123
  - `struct.calcsize` (*function*), 122
  - `struct.pack` (*function*), 122
  - `struct.pack_into` (*function*), 122
  - `struct.unpack` (*function*), 122
  - `struct.unpack_from` (*function*), 122
- `usb` (*module*), 124–138
  - `usb.Bus` (*class*), 125–126
  - `usb.busses` (*function*), 124
  - `usb.Configuration` (*class*), 126–128
  - `usb.Device` (*class*), 128–130
    - `usb.Device.open` (*method*), 129
  - `usb.DeviceHandle` (*class*), 130–134
    - `usb.DeviceHandle.bulkRead` (*method*), 131
    - `usb.DeviceHandle.bulkWrite` (*method*), 131
    - `usb.DeviceHandle.claimInterface` (*method*), 131
    - `usb.DeviceHandle.clearHalt` (*method*), 131
    - `usb.DeviceHandle.controlMsg` (*method*), 131
    - `usb.DeviceHandle.getDescriptor` (*method*), 132
    - `usb.DeviceHandle.getString` (*method*), 132
    - `usb.DeviceHandle.interruptRead` (*method*), 132
    - `usb.DeviceHandle.interruptWrite` (*method*), 133
    - `usb.DeviceHandle.releaseInterface` (*method*), 133
    - `usb.DeviceHandle.reset` (*method*), 133
    - `usb.DeviceHandle.resetEndpoint` (*method*), 133
    - `usb.DeviceHandle.setAltInterface` (*method*), 133
    - `usb.DeviceHandle.setConfiguration` (*method*), 133
  - `usb.Endpoint` (*class*), 134–135
  - `usb.Interface` (*class*), 135–137
  - `usb.USBError` (*class*), 137–138