cpp-delegate Documentation

Release 0.1.post32.dev137914501

Christian Fobel

May 31, 2018

Contents

1 Project Modules **3**
1.1 cpp_delegate Package 3

2 Indices and tables **7**

Python Module Index **9**

Contents:

CHAPTER 1

Project Modules

1.1 cpp_delegate Package

1.1.1 cpp_delegate Package

cpp_delegate.__init__.**dump_cpp_ast**(*env*)

cpp_delegate.__init__.**dump_env**(*env*)

cpp_delegate.__init__.**parse_cpp_ast**(*source*, *env*)

cpp_delegate.__init__.**test**(*v*)

1.1.2 address_of Module

cpp_delegate.address_of.**get_attributes**(*members*)

cpp_delegate.address_of.**render**(*cpp_ast_json*, *attributes*)

1.1.3 context Module

class cpp_delegate.context.**Context**(*cpp_ast_json*, *namespace=''*)

Bases: object

class cpp_delegate.context.**RemoteContext**(*stream*, *cpp_ast_json*, *namespace=''*)

Bases: *cpp_delegate.context.Context*, *cpp_delegate.dir_mixin.DirMixIn*

This class provides access to public variables and fields within a remote context (i.e., namespace).

Variables and fields are accessible as Python attributes, allowing, for example, tab completion in IPython.

Parameters

- **stream** (*serial.Serial*) – A serial connection to the remote device.

- **cpp_ast_json** (*dict*) – A JSON-serializable C++ abstract syntax tree, as parsed by *clang_helpers.clang_ast.parse_cpp_ast(..., format='json')*.
- **namespace** (*str, optional*) – A namespace specifier (e.g., `"foo::bar"`) indicating the namespace to expose.

  A value of `""` corresponds to the top-level namespace.

.<remote variable/field>

An attribute corresponding to each public variable or field within the remote context `namespace`.

_address_of(*label*)

Parameters **label** (*str*) – Name/label of variable or field in remote context.

Returns Address in memory of specified variable or field in remote context.

Return type int

_mem_read(*address*, *size*)

Parameters

- **address** (*int*) – Memory address in remote context.
- **size** (*int*) – Number of bytes to read.

Returns Array of data read from remote context.

Return type np.array(dtype='uint8')

See also:

_read_attribute()

_mem_write(*address*, *data*)

Write data to specified address in remote context.

Parameters

- **address** (*int*) – Memory address in remote context.
- **data** (*numpy.array-like*) – Array or **:module:'numpy'** data type.

See also:

_write_attribute()

_read_attribute(*attr*, **args*)

Parameters

- **attr** (*str*) – Name of attribute in remote context.
- **default** (*object, optional*) – Default return value.

Returns

Value of specified attribute in remote context.

If type is not supported (i.e., not a plain old data type), `default` is returned (if specified).

Return type type of attr

_read_attributes()

Returns

Value of each attribute in remote context.

For each attribute, if type is not supported (i.e., not a plain old data type), value is set to `None`.

Return type dict

See also:

`_write_attribute()`

`_write_attribute`(*attr*, *value*)

Parameters

- **`attr`** (*`str`*) – Name of attribute in remote context.
- **`value`** (*`type of attr`*) – Value to set for specified attribute in remote context.

Raises `TypeError` – If attribute type is not supported (i.e., not a plain old data type).

See also:

`_read_attribute()`

`cpp_delegate.context.`**`get_namespace_path`**(*namespace_str*)

`cpp_delegate.context.`**`get_np_dtype`**(*type_name*, *default=False*)

`cpp_delegate.context.`**`operation_code`**(*v*)

1.1.4 `dir_mixin` Module

class `cpp_delegate.dir_mixin.`**`DirMixIn`**

Bases: `object`

Mix-in to make implementing __dir__ method in subclasses simpler

1.1.5 `member_header` Module

`cpp_delegate.member_header.`**`get_functions`**(*members*)

`cpp_delegate.member_header.`**`render`**(*functions*)

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

cpp_delegate.__init__, 3
cpp_delegate.address_of, 3
cpp_delegate.context, 3
cpp_delegate.dir_mixin, 5
cpp_delegate.member_header, 5

Index

Symbols

_address_of() (cpp_delegate.context.RemoteContext method), 4
_mem_read() (cpp_delegate.context.RemoteContext method), 4
_mem_write() (cpp_delegate.context.RemoteContext method), 4
_read_attribute() (cpp_delegate.context.RemoteContext method), 4
_read_attributes() (cpp_delegate.context.RemoteContext method), 4
_write_attribute() (cpp_delegate.context.RemoteContext method), 5

C

Context (class in cpp_delegate.context), 3
cpp_delegate.__init__ (module), 3
cpp_delegate.address_of (module), 3
cpp_delegate.context (module), 3
cpp_delegate.dir_mixin (module), 5
cpp_delegate.member_header (module), 5

D

DirMixIn (class in cpp_delegate.dir_mixin), 5
dump_cpp_ast() (in module cpp_delegate.__init__), 3
dump_env() (in module cpp_delegate.__init__), 3

G

get_attributes() (in module cpp_delegate.address_of), 3
get_functions() (in module cpp_delegate.member_header), 5
get_namespace_path() (in module cpp_delegate.context), 5
get_np_dtype() (in module cpp_delegate.context), 5

O

operation_code() (in module cpp_delegate.context), 5

P

parse_cpp_ast() (in module cpp_delegate.__init__), 3

R

RemoteContext (class in cpp_delegate.context), 3
render() (in module cpp_delegate.address_of), 3
render() (in module cpp_delegate.member_header), 5

T

test() (in module cpp_delegate.__init__), 3