Masterarbeit Grosch Documentation

Release 0.4c0

Tobias Grosch

May 29, 2015

Contents:

CHAPTER

ONE

BACNET PACKAGE

1.1 Subpackages

1.1.1 bacnet.api package

Module contents

BACnet API Module

This module provides basic api functionality.

class bacnet.api.**BACnetAPI**(**args*)
Bases: object

This class unifies all API accessible functions.

create(*line*)
This function creates a request for the specified command.

Parameters **line** – command line

Returns request object

static parse(*apdu*)
This function returns a parsed dictionary from provided message. :param apdu: message :return: parsed dict

receive(*block=True*, *timeout=None*)
This function checks for available incoming requests

Parameters

- **block** – block
- **timeout** – timeout

Returns incoming response or None

send(*request*, *block=True*, *timeout=None*)
This function transmits the request.

Parameters

- **request** – outgoing request
- **block** – block
- **timeout** – timeout

Returns request was queued

1.1.2 bacnet.app package

Subpackages

bacnet.app.handler package

Submodules

bacnet.app.handler.error module

Error Module This module provides error handlers.

bacnet.app.handler.error.**do_AbortPDU**(*self*, *apdu*)
This function handles abortions.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.error.**do_Error**(*self*, *apdu*)
This function handles rejections.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.error.**do_RejectPDU**(*self*, *apdu*)
This function handles rejections.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.events module

Event Handler Module This module provides event handlers.

bacnet.app.handler.events.**do_ConfirmedCOVNotificationRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.events.**do_ConfirmedEventNotificationRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.events.**do_GetEventInformationRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.events.**do_SubscribeCOVPropertyRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.events.**do_SubscribeCOVRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.events.**do_UnconfirmedCOVNotificationRequest**(*self*, *apdu*)
This function reads data from request.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.events.**do_UnconfirmedEventNotificationRequest**(*self*, *apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.fileaccess module

File Access Handler Module This module provides file access handlers.

bacnet.app.handler.fileaccess.**do_AtomicReadFileRequest**(*self*, *apdu*)
This function reads data from read file request.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.fileaccess.**do_AtomicWriteFileRequest**(*self*, *apdu*)
This function reads data from write file request.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.objects module

Objects Handler Module This module provides object handlers.

bacnet.app.handler.objects.**do_CreateObjectRequest**(*self*, *apdu*)
This functions handles object creation.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.objects.**do_DeleteObjectRequest**(*self*, *apdu*)
This functions handles object deletion.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple module

Application Handler Module This module provides basic functions for value casting, property creation and general handlers.

bacnet.app.handler.simple.**do_IAmRequest**(*self*, *apdu*)
This function reads data from I Am request.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**do_IHaveRequest**(*self*, *apdu*)
This function reads data from I Have request.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**do_ReadPropertyMultipleRequest**(*self*, *apdu*)
This function reads all values from request.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.simple.**do_ReadPropertyRequest**(*self*, *apdu*)
This function reads value from request response.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**do_WhoHasRequest**(*self*, *apdu*)
This function responses to WhoHas requests.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**do_WhoIsRequest**(*self*, *apdu*)
This function responses to WhoIs requests.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**do_WritePropertyMultipleRequest**(*self*, *apdu*)
This function reads all values from request.

Parameters **apdu** – incoming message

Returns response

bacnet.app.handler.simple.**do_WritePropertyRequest**(*self*, *apdu*)
This function reads value from request response.

Parameters **apdu** – incoming message

Returns None

bacnet.app.handler.simple.**property_to_result**(*obj*, *prop_id*, *prop_index=None*)
This function creates an appropriate ReadAccessResultElement.

Parameters

- **obj** – object

- **prop_id** – property id
- **prop_index** – property array index

Returns ReadAccessResultElement instance

bacnet.app.handler.simple.**read_property_any**(*obj*, *prop_id*, *prop_index=None*, *prop=None*)

This function creates an appropriate property value of type Any.

Parameters

- **obj** – object
- **prop_id** – property id
- **prop_index** – property array index
- **prop** – property instance

Returns Any instance

Module contents

Application Handler Module This module provides handler for communication callbacks.

class bacnet.app.handler.**HandlerApplication**(**args*, ***kwargs*)

Bases: bacpypes.app.Application

This class provides handler functionality for basic requests.

add_cov_subscription(*apdu*, *obj*, *prop=None*)

This function adds subscription to cov subscription list.

Parameters

- **apdu** – incoming message
- **obj** – object
- **prop** – property

Returns None

check_remote_subscription_updates(*apdu*)

This function checks if the cov notification was initiated correctly.

Parameters **apdu** – incoming message

Returns remote subscription exists

check_subscription_updates(**args*, ***kwargs*)

This function is the actual wrapper of the function call.

delete_cov_subscription(**args*, ***kwargs*)

This function is the actual wrapper of the function call.

delete_cov_subscriptions(**args*, ***kwargs*)

This function is the actual wrapper of the function call.

do_AbortPDU(*apdu*)

This function handles abortions.

Parameters **apdu** – incoming message

Returns None

do_AtomicReadFileRequest(*apdu*)
This function reads data from read file request.

Parameters **apdu** – incoming message

Returns response

do_AtomicWriteFileRequest(*apdu*)
This function reads data from write file request.

Parameters **apdu** – incoming message

Returns response

do_ConfirmedCOVNotificationRequest(*apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

do_CreateObjectRequest(*apdu*)
This functions handles object creation.

Parameters **apdu** – incoming message

Returns None

do_DeleteObjectRequest(*apdu*)
This functions handles object deletion.

Parameters **apdu** – incoming message

Returns None

do_Error(*apdu*)
This function handles rejections.

Parameters **apdu** – incoming message

Returns None

do_IAmRequest(*apdu*)
This function reads data from I Am request.

Parameters **apdu** – incoming message

Returns None

do_IHaveRequest(*apdu*)
This function reads data from I Have request.

Parameters **apdu** – incoming message

Returns None

do_ReadPropertyMultipleRequest(*apdu*)
This function reads all values from request.

Parameters **apdu** – incoming message

Returns response

do_ReadPropertyRequest(*apdu*)
This function reads value from request response.

Parameters **apdu** – incoming message

Returns None

do_RejectPDU(*apdu*)
This function handles rejections.

Parameters **apdu** – incoming message

Returns None

do_SubscribeCOVPropertyRequest(*apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

do_SubscribeCOVRequest(*apdu*)
This function reads data from request and returns response.

Parameters **apdu** – incoming message

Returns response

do_UnconfirmedCOVNotificationRequest(*apdu*)
This function reads data from request.

Parameters **apdu** – incoming message

Returns None

do_WhoHasRequest(*apdu*)
This function responses to WhoHas requests.

Parameters **apdu** – incoming message

Returns None

do_WhoIsRequest(*apdu*)
This function responses to WhoIs requests.

Parameters **apdu** – incoming message

Returns None

do_WritePropertyMultipleRequest(*apdu*)
This function reads all values from request.

Parameters **apdu** – incoming message

Returns response

do_WritePropertyRequest(*apdu*)
This function reads value from request response.

Parameters **apdu** – incoming message

Returns None

do_indication(*apdu*)
This function initiates appropriate function call for a received request.

Parameters **apdu** – incoming message

Returns None

get_device(*address*)
This function returns device identifier if known.

Parameters **address** – device address

Returns device identifier

get_object_by_id(*obj_id*)
This function returns object by id.

Parameters **obj_id** – object id

Returns object

get_object_by_name(*obj_name*)
This function returns object by name.

Parameters **obj_name** – object name

Returns object

get_object_id(*obj_type*)
This function provides a unique object id.

Parameters **obj_type** – object type

Returns object id

handle_remote_subscription(*apdu*)
This function stores a new remote subscription.

Parameters **apdu** – outgoing request

Returns None

receive_remote_notification(*apdu*)
This function checks if the cov notification was initiated correctly.

Parameters **apdu** – incoming message

Returns remote subscription exists

renew_cov_subscription(**args*, ***kwargs*)
This function is the actual wrapper of the function call.

send_cov_notification(*subscription*, *values*)
This function transmits cov notifications for subscriptions

Parameters

- **subscription** – subscription
- **values** – sequence of property values

Returns None

bacnet.app.handler.**lock_subscriptions**(*func=None*, *block=True*, *retries=3*, *sleep=0.05*)
This function is a decorator for functions to lock subscriptions during execution.

Parameters

- **func** – function reference
- **block** – wait until lock is acquired
- **retries** – retry count if not block
- **sleep** – sleep time in secs if not block

Returns updated function

bacnet.app.handler.**restart_on_failure**(*func=None*, *retries=3*, *fail_time=1*)
This function is a decorator for thread functions to restart after failure :param func: function reference :param retries: restart count within fail time :param fail_time: time between failures to count retries :return: updated function

Submodules

bacnet.app.basic module

Application Basic Module

This module provides a basic BACpypes application object.

class bacnet.app.basic.**BasicApplication**(**args*, ***kwargs*)
Bases: *bacnet.app.handler.HandlerApplication*, multiprocessing.process.Process

This class describes a basic BACpypes application including simple functionality.

add_object(*obj*)
This functions adds an object to the collection.

Parameters **obj** – object

Returns None

check_indication_queue(**args*, ***kwargs*)
This function is the actual wrapper of the function call.

check_queue(**args*, ***kwargs*)
This function is the actual wrapper of the function call.

confirmation(**args*, ***kwargs*)
This function queues incoming messages.

Parameters **apdu** – incoming message

Returns None

console_interrupt(**args*)
This function is the catch for interrupts.

Returns None

delete_object(*obj*)
This function removes object from collection.

Parameters **obj** – object

Returns None

indication(*apdu*)
This function queues received requests.

Parameters **apdu** – incoming message

Returns None

iter_objects()
This function returns an iterator of the object collection.

Returns object iterator

request (**args*, ***kwargs*)
This function handles outgoing messages.

Parameters **apdu** – outgoing message

Returns None

response (**args*, ***kwargs*)
This function handles outgoing messages as response to a request.

Parameters **apdu** – outgoing message

Returns

run (**args*, ***kwargs*)
This function is the actual wrapper of the function call.

shutdown (**args*)
This function shuts down the process.

Returns None

stdout = None

terminate ()
This function terminates processing.

Returns None

update_devices (**args*, ***kwargs*)
This function is the actual wrapper of the function call.

bacnet.app.basic.**poll_hardware** (*application*, *obj_tuple*)
This function polls hardware values.

Parameters

- **application** – application object
- **obj_tuple** – object tuples

Returns None

bacnet.app.define module

Application Define Module

This module contains the user specific definition of a BACpypes Application and Console Commands.

class bacnet.app.define.**Application** (**args*, ***kwargs*)
Bases: *bacnet.app.basic.BasicApplication*

This class describes the user specific Application functionality including basic builtin functions like “whois” and “iam”.

Module contents

Application Module

This module contains just a single public function for creating a basic device and application to ensure BACnet conform communications by using BACpypes.

Function for creating device and application: create_device_and_app(args)

bacnet.app.**create_app**(*args*, *device_init=None*, *stdout=None*, ***kwargs*)

This function creates two BACpypes objects needed for BACnet communications. This includes a local device for identification of the BACnet device and grouping of related BACnet objects as well as an application for message handling from the network.

Parameters **args** – configuration arguments

Returns device, application

1.1.3 bacnet.console package

Subpackages

bacnet.console.creator package

Submodules

bacnet.console.creator.events module

Event Handler Module This module provides COV and event requests.

bacnet.console.creator.events.**subscribe_request**(*args*, *console=None*)

This function creates a write file stream request.

Usage: subscribe <address> <lifetime> <confirmed> <type> <instance> [<property> <increment> [<index>]]

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.fileaccess module

File Access Handler Module This module provides file access requests.

bacnet.console.creator.fileaccess.**rdrec_request**(*args*, *console=None*)

This function creates a read file record request.

Usage: rdrec <address> <instance> <start> <count>

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.fileaccess.**rdstr_request**(*args*, *console=None*)

This function creates a read file stream request.

Usage: rdstr <address> <instance> <start> <count>

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.fileaccess.**wrrec_request**(*args*, *console=None*)

This function creates a write file record request.

Usage: wrrec <address> <instance> <start> <count> (<data>)* ...

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.fileaccess.**wrstr_request**(*args*, *console=None*)

This function creates a write file stream request.

Usage: wrstr <address> <instance> <start> <data>

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.local module

Local Handler Module This module provides local access requests. - DEPRECATED !

bacnet.console.creator.local.**local_list**(*args*, *i*, *obj=None*, *objs=None*)

This function lists all objects within the application, all properties within an object, or the property value of a property.

Parameters

- **args** – list of parameters
- **obj** – object

Returns None

bacnet.console.creator.local.**local_set**(*args*, *i*, *obj*)

This function sets the property value of a local property.

Parameters

- **args** – list of parameters
- **obj** – object

Returns None

bacnet.console.creator.local.**print_values**(*values*)

This function prints all arguments.

Parameters **values** – list of argument tuples

Returns None

bacnet.console.creator.objects module

Objects Handler Module This module provides object requests.

bacnet.console.creator.objects.**create_request**(*args*, *console=None*)
This function creates a create object request.

Usage: create <address> <type> [<vendor>] (<property> <value> [<index> [priority]])* ...

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.objects.**delete_request**(*args*, *console=None*)
This function creates a delete object request.

Usage: delete <address> <type> <instance>

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple module

Simple Handler Module This module contains rudimentary request functions.

bacnet.console.creator.simple.**cast_value**(*value*, *data_type*, *prop_index=None*, *prop=None*)
This function casts value to data type.

Parameters

- **value** – object
- **data_type** – data type
- **prop_index** – property array index
- **prop** – property instance

Returns casted value

bacnet.console.creator.simple.**iam_request**(*args*, *console=None*)
This function creates a iam request.

Usage: iam [<address>]

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple.**ihave_request**(*args*, *console=None*)
This function creates a ihave request.

Usage: ihave [<address>] (<name> | <type> <instance>)

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple.**read_request**(*args*, *console=None*)
This function creates a read request for multiple objects.

Usage: read <address> (<type> <instance> (<property> [<index>])+ ...)+ ...

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple.**whohas_request**(*args*, *console=None*)
This function creates a whohas request.

Usage: whohas [<address>] (<name> | <type> <instance>) [<lowlimit> <highlimit>]

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple.**whois_request**(*args*, *console=None*)
This function creates a whois request.

Usage: whois [<address>] [<lowlimit> <highlimit>]

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

bacnet.console.creator.simple.**write_request**(*args*, *console=None*)
This function creates a write request for multiple objects.

Usage: write <address> (<type> <instance> (<property> <value> [<index> [<priority>]])+ ...)+ ...

Parameters

- **args** – list of parameters
- **console** – console object

Returns request

Module contents

Console Handler Module This module provides requests and parser.

bacnet.console.creator.**request_creator**(*line*, *console=None*, *local_id=1*)
This function parses arguments and calls specific command.

Parameters

- **line** – string of command line
- **console** – console object

Returns request object

bacnet.console.parser package

Submodules

bacnet.console.parser.events module

Event Parser Module This module provides event parsers.

bacnet.console.parser.events.**confirmed_cov_notification_request**(*apdu*)
This function parses confirmed cov notification request.

Parameters **apdu** – ConfirmedCOVNotificationRequest

Returns parsed data, output

bacnet.console.parser.events.**subscribe_cov_property_request**(*apdu*)
This function parses subscribe cov property request.

Parameters **apdu** – SubscribeCOVPropertyRequest

Returns parsed data, output

bacnet.console.parser.events.**subscribe_cov_request**(*apdu*)
This function parses subscribe cov request.

Parameters **apdu** – SubscribeCOVRequest

Returns parsed data, output

bacnet.console.parser.events.**unconfirmed_cov_notification_request**(*apdu*)
This function parses unconfirmed cov notification request.

Parameters **apdu** – UnconfirmedCOVNotificationRequest

Returns parsed data, output

bacnet.console.parser.fileaccess module

File Access Parser Module This module provides file access parsers.

bacnet.console.parser.fileaccess.**read_file_ack**(*apdu*)
This function parses atomic read file acknowledgements.

Parameters **apdu** – AtomicReadFileACK

Returns parsed data, output

bacnet.console.parser.fileaccess.**write_file_ack**(*apdu*)
This function parses atomic write file acknowledgements.

Parameters **apdu** – AtomicWriteFileACK

Returns parsed data, output

bacnet.console.parser.simple module

Simple Parser Module This module contains rudimentary parser functions.

bacnet.console.parser.simple.**iam_request**(*apdu*)
This function parses i am requests.

Parameters **apdu** – IAmRequest

Returns parsed data, output

bacnet.console.parser.simple.**ihave_request**(*apdu*)
This function parses i have requests.

Parameters **apdu** – IHaveRequest

Returns parsed data, output

bacnet.console.parser.simple.**read_property_ack**(*apdu*)
This function parses read property acknowledgements.

Parameters **apdu** – ReadPropertyACK

Returns parsed data, output

bacnet.console.parser.simple.**read_property_multiple_ack**(*apdu*)
This function parses read property multiple acknowledgements.

Parameters **apdu** – ReadPropertyMultipleACK

Returns parsed data, output

bacnet.console.parser.simple.**whohas_request**(*apdu*)
This function parses who has requests.

Parameters **apdu** – WhoHasRequest

Returns parsed data, output

bacnet.console.parser.simple.**whois_request**(*apdu*)
This function parses who is requests.

Parameters **apdu** – WhoIsRequest

Returns parsed data, output

Module contents

Acknowledgement Parser Module This module provides acknowledgement parsing handlers.

bacnet.console.parser.**response_parser**(*apdu*, *console=None*)
This function parses request messages and returns parsed data.

Parameters

- **apdu** – request message

- **console** – console object

Returns parsed data

Submodules

bacnet.console.basic module

Console Basic Module

This module provides a basic BACpypes console commands object.

class bacnet.console.basic.**BasicConsole**(*device*, *application*, **args*, ***kwargs*)
Bases: cmd.Cmd, multiprocessing.process.Process

This class describes a basic BACpypes shell including simple commands.

application = None

callback(*apdu*)
This function provides a callback for command executions.

Parameters **apdu** – message

Returns None

check_queue()
This function checks for queued outgoing requests.

Returns None

console_interrupt(**args*)
This function is the catch for interrupts.

Returns None

default(*line*)
This function handles the command line if no command was found.

Parameters **line** – command line

Returns None

device = None

do_EOF(*args*)
This function provides an exit.

Usage: EOF

Parameters **args** – string of parameters

Returns None

do_buggers(*args*)
This function lists all buggers.

Usage: buggers [<prefix>]

Parameters **args** – string of parameters

Returns None

do_bugin (*args*)
This function activates debugging for mentioned buggers.

Usage: bugin (<bugger>)+

Parameters **args** – string of parameters

Returns None

do_bugout (*args*)
This function deactivates debugging for mentioned buggers.

Usage: bugout (<bugger>)+

Parameters **args** – string of parameters

Returns None

do_exit (*args*)
This function provides an exit.

Usage: exit

Parameters **args** – string of parameters

Returns None

do_gc (*args*)
This function prints current garbage collection information.

Usage: gc

Parameters **args** – string of parameters

Returns None

do_info (*args*)
This function adjusts package output.

Usage: info [(on | off) (<msg_type>)*]

Parameters **args** – string of parameters

Returns None

do_shell (*args*)
This function provides an access to shell execution.

Usage: ! <command>

Parameters **args** – string of parameters

Returns None

do_version (*args*)
This function prints current software version.

Usage: version

Parameters **args** – string of parameters

Returns None

emptyline ()
This function ignores empty lines.

Returns None

get_names()
This function collects all attributes.

Returns list of attributes

help_create(*text='Usage: create <address> <type> [<vendor>] (<property> <value> [<index> [priority]])* ...'*)

help_delete(*text='Usage: delete <address> <type> <instance>'*)

help_iam(*text='Usage: iam [<address>]'*)

help_ihave(*text='Usage: ihave [<address>] (<name> | <type> <instance>)'*)

help_rdrec(*text='Usage: rdrec <address> <instance> <start> <count>'*)

help_rdstr(*text='Usage: rdstr <address> <instance> <start> <count>'*)

help_read(*text='Usage: read <address> (<type> <instance> (<property> [<index>])+ ...)+ ...'*)

help_subscribe(*text='Usage: subscribe <address> <lifetime> <confirmed> <type> <instance> [<property> <increment> [<index>]]'*)

help_whohas(*text='Usage: whohas [<address>] (<name> | <type> <instance>) [<lowlimit> <highlimit>]'*)

help_whois(*text='Usage: whois [<address>] [<lowlimit> <highlimit>]'*)

help_write(*text='Usage: write <address> (<type> <instance> (<property> <value> [<index> [<priority>]])+ ...)+ ...'*)

help_wrrec(*text='Usage: wrrec <address> <instance> <start> <count> (<data>)* ...'*)

help_wrstr(*text='Usage: wrstr <address> <instance> <start> <data>'*)

onecmd(*line*, *queue=True*)
This function interprets the command line.

Parameters **line** – user command line

Returns command result

postloop()
This function wraps up readline history after loop has finished.

Returns None

preloop()
This function prepares loop by loading readline history file.

Returns None

print_text(*text*)
This function prints help text.

Parameters **text** – help text

Returns None

run()
This function initiates processing.

Returns None

shutdown(**args*)
This function shuts down the process.

Returns None

terminate()
This function terminates processing.

Returns None

bacnet.console.define module

Console Define Module

This module is an empty shell.

class bacnet.console.define.**Console**(*device*, *application*, **args*, ***kwargs*)
Bases: *bacnet.console.basic.BasicConsole*

This Class describes the user specific Console Commands including basic builtin functions like "read" and "write".

Module contents

Console Module

This module contains just a single public function for creating an object for a set of console commands to ensure BACnet conform communications by using BACpypes.

Function for creating console commands: create_console(application)

bacnet.console.**create_console**(*device*, *application*, ***kwargs*)
This function creates a single BACpypes object to support a shell to control BACnet messages.

Parameters

- **device** – device object
- **application** – application object

Returns console object

1.1.4 bacnet.debugging package

Submodules

bacnet.debugging.log module

Debugging Log Module

This module contains necessary objects to support logging in multiprocessing environments.

class bacnet.debugging.log.**LoggingFormatter**(*fmt=None*, *color=None*)
Bases: logging.Formatter

This class is a wrapper for logging formatter to support coloring.

format(*record*)
This function formats the message.

Parameters **record** – logging record

Returns message text

class bacnet.debugging.log.**MultiProcessingLog**(*name*, *mode='a'*, *maxsize=0*, *backupcount=0*, ***kwargs*)

Bases: logging.Handler

This class sends logs to queues.

close()

This function closes the handler.

Returns None

emit(*record*)

This function sends the message to queue.

Parameters **record** – message

Returns None

receive()

This function polls for received logs.

Returns None

send(*record*)

This function queues the message.

Parameters **record** – message

Returns None

setFormatter(*fmt*)

This function sets the formatter.

Parameters **fmt** – formatter instance

Returns

Module contents

Debugging Module

This module contains basic debugging functionality.

bacnet.debugging.**ModuleLogger**(*context=None*, *details=0*, *level=None*, *module_name=None*, *formatter=True*)

This function creates a general logger.

Returns None

class bacnet.debugging.**TimeoutJoinableQueue**(**args*, ***kwargs*)

Bases: multiprocessing.queues.JoinableQueue

This class is an updated version of JoinableQueue with join timeout.

join(*timeout=None*)

This function overrides the join function with timeout support.

bacnet.debugging.**bacnet_debug**(*func=None*, *formatter=True*, *level=None*)

This function is a decorator for objects to add debugging methods.

Parameters

- **formatter** – set formatter string

- **level** – set log level

Returns updated object

bacnet.debugging.**get_formatter**(*color=None*)
This function returns a logging formatter instance.

Parameters **color** – set color

Returns logging formatter

bacnet.debugging.**get_loggers**(*prefix=None*)
This function returns the list of all loggers.

Returns list of loggers

bacnet.debugging.**iso_now**()
This function returns current time in iso format.

Returns time in iso format

bacnet.debugging.**set_debug**(*args=None*, *stream=None*)
This function sets basic logging parameters.

Parameters **args** – arguments

Returns log queue

bacnet.debugging.**set_handler**(*name*, ***kwargs*)
This function assigns logging handler to certain loggers.

Parameters **name** – logger name

Returns None

1.1.5 bacnet.object package

Subpackages

bacnet.object.basic package

Submodules

bacnet.object.basic.cov_support module

COV Support Module This module contains a list of all objects and properties supporting COV events.

bacnet.object.basic.device module

Object Basic Module This module contains basic definitions of BACpypes Object.

class bacnet.object.basic.device.**LocalDeviceObject**(***kwargs*)
Bases: *bacnet.object.basic.general.DeviceObject*

This class provides additional functionality for device objects.

defaultProperties = {'numberOfApduRetries': 3, 'maxApduLengthAccepted': 1024, 'apduTimeout': 3000, 'apduSe

properties = [<bacnet.object.properties.CurrentTimeProperty 'localTime'>, <bacnet.object.properties.CurrentDatePr

bacnet.object.basic.fileaccess module

File Access Object Module This module contains File Access Objects.

class bacnet.object.basic.fileaccess.**RecordAccessFileObject**(***kwargs*)
Bases: *bacnet.object.basic.general.FileObject*

This class provides read/write access to file records.

ReadFile(*start*, *count*)
This function provides read access.

Parameters

- **start** – start record
- **count** – record count

Returns record data

WriteFile(*start*, *count*, *data*)
This function provides write access.

Parameters

- **start** – start record
- **count** – record count
- **data** – record data

Returns start record

class bacnet.object.basic.fileaccess.**StreamAccessFileObject**(***kwargs*)
Bases: *bacnet.object.basic.general.FileObject*

This class provides read/write access to file streams.

ReadFile(*start*, *count*)
This function provides read access.

Parameters

- **start** – start position
- **count** – letter count

Returns record data

WriteFile(*start*, *data*)
This function provides write access.

Parameters

- **start** – start position
- **data** – data

Returns start position

get_filename()
This function returns the absolute path and file name.

Returns file name

lock_file()
This function locks the file.

Returns None

unlock_file()
This function unlocks the file.

Returns None

bacnet.object.basic.general module

Object Basic Module This module contains basic definitions of BACpypes Object.

class bacnet.object.basic.general.**AccessCredentialObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessCredentialObject.

objectType = 'accessCredential'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccessDoorObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessDoorObject.

objectType = 'accessDoor'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccessPointObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessPointObject.

objectType = 'accessPoint'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccessRightsObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessRightsObject.

objectType = 'accessRights'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccessUserObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessUserObject.

objectType = 'accessUser'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccessZoneObject**(**kwargs)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccessZoneObject.

objectType = 'accessZone'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AccumulatorObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AccumulatorObject.

objectType = 'accumulator'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AnalogInputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AnalogInputObject.

objectType = 'analogInput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AnalogOutputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AnalogOutputObject.

objectType = 'analogOutput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AnalogValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AnalogValueObject.

objectType = 'analogValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**AveragingObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes AveragingObject.

objectType = 'averaging'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**BinaryInputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes BinaryInputObject.

objectType = 'binaryInput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**BinaryOutputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes BinaryOutputObject.

objectType = 'binaryOutput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**BinaryValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes BinaryValueObject.

objectType = 'binaryValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**BitStringValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes BitStringValueObject.

objectType = 'bitstringValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**CalendarObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes CalendarObject.

objectType = 'calendar'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**CharacterStringValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes CharacterStringValueObject.

objectType = 'characterstringValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**CommandObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes CommandObject.

objectType = 'command'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**CredentialDataInputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes CredentialDataInputObject.

objectType = 'credentialDataInput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**DatePatternValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes DatePatternValueObject.

objectType = 'datePatternValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**DateTimePatternValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes DateTimePatternValueObject.

objectType = 'datetimePatternValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**DateTimeValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes DateTimeValueObject.

objectType = 'datetimeValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**DateValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes DateValueObject.

objectType = 'dateValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**DeviceObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes DeviceObject.

objectType = 'device'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**EventEnrollmentObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes EventEnrollmentObject.

objectType = 'eventEnrollment'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**EventLogObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes EventLogObject.

objectType = 'eventLog'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**FileObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes FileObject.

objectType = 'file'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**GlobalGroupObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes GlobalGroupObject.

objectType = 'globalGroup'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**GroupObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes GroupObject.

objectType = 'group'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**IntegerValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes IntegerValueObject.

objectType = 'integerValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**LargeAnalogValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes LargeAnalogValueObject.

objectType = 'largeAnalogValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**LifeSafetyPointObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes LifeSafetyPointObject.

objectType = 'lifeSafetyPoint'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**LifeSafetyZoneObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes LifeSafetyZoneObject.

objectType = 'lifeSafetyZone'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**LoadControlObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes LoadControlObject.

objectType = 'loadControl'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**LoopObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes LoopObject.

objectType = 'loop'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**MultiStateInputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes MultiStateInputObject.

objectType = 'multiStateInput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**MultiStateOutputObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes MultiStateOutputObject.

objectType = 'multiStateOutput'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**MultiStateValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes MultiStateValueObject.

objectType = 'multiStateValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**NetworkSecurityObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes NetworkSecurityObject.

objectType = 'networkSecurity'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**NotificationClassObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes NotificationClassObject.

objectType = 'notificationClass'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**Object**(***kwargs*)
Bases: bacpypes.object.Object

This class is an extension of the bacpypes Object class to support COV notifications.

ReadProperty(*propid*, *arrayIndex=None*, ***kwargs*)
This function handles reading of property values.

Parameters

- **propid** – property identifier
- **arrayIndex** – property index

Returns property value

WriteProperty(*propid*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*, ***kwargs*)
This function handles writing of property values.

Parameters

- **propid** – property identifier
- **value** – property value
- **arrayIndex** – property index
- **priority** – write priority
- **direct** – direct write

Returns property value

add_cov_subscription(*subscription*)
This function adds subscription to cov subscription list.

Parameters **subscription** – subscription

Returns None

cov_supported(*prop=None*)
This function checks if COV subscription is supported.

Parameters **prop** – property object or identifier

Returns COV is supported

delete_cov_subscription(*subscriptions*, *inform_app=False*)
This function deletes subscription from cov subscription list.

Parameters

- **subscriptions** – subscriptions
- **inform_app** – inform application

Returns None

get_property(*identifier*)
This function returns the property for identifier. :param identifier: property identifier :return: property

get_value(*identifier*)
This function returns the value for property identifier. :param identifier: property identifier :return: value

poll_hardware()
This function checks for value updates for hardware.

Returns None

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

renew_cov_subscription(*subscription*)
This function checks if subscription can be renewed and returns boolean of this action.

Parameters **subscription** – subscription

Returns subscription renewed

set_application(*application*)
This function defines the application associated to this object.

Parameters **application** – application object

Returns None

set_value(*identifier*, *value*)
This function sets the value for property identifier. :param identifier: property identifier :param value: property value :return: None

class bacnet.object.basic.general.**OctetStringValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes OctetStringValueObject.

objectType = 'octetstringValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**PositiveIntegerValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes PositiveIntegerValueObject.

objectType = 'positiveIntegerValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**ProgramObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes ProgramObject.

objectType = 'program'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**PulseConverterObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes PulseConverterObject.

objectType = 'pulseConverter'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**ScheduleObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes ScheduleObject.

objectType = 'schedule'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**StructuredViewObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes StructuredViewObject.

objectType = 'structuredView'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**TimePatternValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes TimePatternValueObject.

objectType = 'timePatternValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**TimeValueObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes TimeValueObject.

objectType = 'timeValue'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**TrendLogMultipleObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes TrendLogMultipleObject.

objectType = 'trendLogMultiple'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

class bacnet.object.basic.general.**TrendLogObject**(***kwargs*)
Bases: *bacnet.object.basic.general.Object*

This class is an cov extended version of bacpypes TrendLogObject.

objectType = 'trendLog'

properties = [<bacnet.object.properties.ObjectIdentifierProperty 'objectIdentifier'>, <bacnet.object.properties.Reada

bacnet.object.basic.general.**cls_obj**
alias of *Object*

bacnet.object.basic.general.**get_datatype**(*cls*, *prop_id*, *vendor_id=0*)
This function returns the data type for a specified class and property identifier.

Parameters

- **cls** – object class
- **prop_id** – property identifier

Returns object type

bacnet.object.basic.general.**get_object_class**(*cls*, *vendor_id=0*)
This function returns the object type for a specified class.

Parameters **cls** – object class

Returns object type

bacnet.object.basic.general.**new_class**
alias of *TrendLogMultipleObject*

bacnet.object.basic.general.**new_property**(*cls_type*, *old_prop*)
This function converts bacpypes properties into new bacnet properties.

Parameters

- **cls_type** – object type
- **old_prop** – old property

Returns new property

bacnet.object.basic.general.**register_object_type**(*cls*, *vendor_id=0*)
This function stores new object types.

Parameters **cls** – object class

Returns None

bacnet.object.basic.program module

Object Basic Module This module contains basic definitions of BACpypes Object.

class bacnet.object.basic.program.**ExecProgramObject**(***kwargs*)
Bases: *bacnet.object.basic.general.ProgramObject*

This class provides additional functionality for python programs.

WriteProperty(*prop_id*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function writes to property.

Parameters

- **prop_id** – property identifier
- **value** – value
- **arrayIndex** – index
- **priority** – priority

- **direct** – direct

Returns None

do_haltProgram()
This functions watches over the halt process of the executed program.

Returns None

haltProgram()
This function halts the program.

Returns None

loadProgram()
This function loads the program.

Returns None

runProgram()
This function runs the program.

Returns None

safe_delete()
This function handles safe deletion

Returns None

set_application(*application*)
This function defines the application associated to this object.

Parameters **application** – application object

Returns None

unloadProgram()
This function unloads the program.

Returns None

watchdog(**funcs*)
This function watches over the executed program changes.

Returns None

Module contents

Object Basic Module This module contains basic definitions of BACpypes Object.

bacnet.object.hardware package

Subpackages

bacnet.object.hardware.terra package

Submodules

bacnet.object.hardware.terra.ablib_dummy module

Terra_Board ABLIB_Dummy Module This module contains dummy classes for hardware abstraction.

class bacnet.object.hardware.terra.ablib_dummy.**ABLIB_Dummy**(**args*, ***kwargs*)
Bases: object

This class is a generic dummy class.

identifier = None

name = ''

class bacnet.object.hardware.terra.ablib_dummy.**BinaryReadDummy**(**args*, ***kwargs*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.ABLIB_Dummy*

This class is a dummy for reading binary values.

get(*identifier=None*)
This function is a dummy for getting the led status.

value = False

class bacnet.object.hardware.terra.ablib_dummy.**BinaryWriteDummy**(**args*, ***kwargs*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.BinaryReadDummy*

This class is a dummy for reading and writing binary values

off(*identifier=None*)
This function is a dummy for turning off the led.

on(*identifier=None*)
This function is a dummy for turning on the led.

class bacnet.object.hardware.terra.ablib_dummy.**Daisy11**(*connector_id*, *led_id*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.BinaryWriteDummy*

This class is a dummy for the terra board's led array.

name = 'LED'

class bacnet.object.hardware.terra.ablib_dummy.**Daisy19**(*connector_id*, *position*, *output_id*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.BinaryWriteDummy*

This class is a dummy for the terra board's 4 channel output switch.

name = 'Output'

class bacnet.object.hardware.terra.ablib_dummy.**Daisy20**(**args*, ***kwargs*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.BinaryReadDummy*

This class is a dummy for the terra board's adc.

convert(*value*)
This function is a dummy to convert the bit value into voltage.

Parameters **value** – bit value

Returns voltage

name = 'ADC'

class bacnet.object.hardware.terra.ablib_dummy.**Daisy22**(**args*, ***kwargs*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.BinaryWriteDummy*

This class is a dummy for the terra board's backlight led of the lcd display.

name = 'Backlight LED'

class bacnet.object.hardware.terra.ablib_dummy.**Daisy24**(**args*, ***kwargs*)
Bases: *bacnet.object.hardware.terra.ablib_dummy.ABLIB_Dummy*

This class is a dummy for the terra board's lcd display and button array.

backled = <bacnet.object.hardware.terra.ablib_dummy.Daisy22 'Backlight LED None'>

name = 'LCD'

pressed(*keyid*)
This function is a dummy for checking pressed buttons.

Parameters **keyid** – button id

Returns button pressed

putstring(*value*)
This function is a dummy for putting a string to the LCD display.

Parameters **value** – string

Returns None

setcurpos(*x*, *y*)
This function is a dummy for setting the cursor on the LCD display.

Parameters

- **x** – x position
- **y** – y position

Returns None

Module contents

Terra Module This module contains Terra Board specific objects.

class bacnet.object.hardware.terra.**AbstractDaisy20**(*identifier*)
Bases: object

This class is an abstraction of the ablib.Daisy20 class.

get()
This function is a wrapper for the get function.

Returns voltage

identifier = None

class bacnet.object.hardware.terra.**AbstractDaisy24Backlight**(*identifier=None*)
Bases: object

This class is an abstraction for backlight of the ablib.Daisy24 class.

get()
This function is a wrapper for the backled.get function.

Returns led on/off

off()
This function is a wrapper for the backled.off function.

Returns None

on()
This function is a wrapper for the backled.on function.

Returns None

class bacnet.object.hardware.terra.**AbstractDaisy24Button**(*identifier*)
Bases: object

This class is an abstraction for buttons of the ablib.Daisy24 class.

get()
This function is a wrapper for the pressed function.

Returns button pressed

identifier = None

class bacnet.object.hardware.terra.**AbstractDaisy24LCD**(*identifier=None*)
Bases: object

This class is an abstraction for lcd of the ablib.Daisy24 class.

get()
This function is a providing a value buffer.

Returns button pressed/not pressed

identifier = None

value = ''

write(*value*)
This function is a wrapper for the putstring and setcurpos function

Returns None

class bacnet.object.hardware.terra.**New_Daisy20**(*max_voltage=10*, *bits=10*)
Bases: object

DAISY-20 (ADC module)

adc_path = '/sys/bus/iio/devices/iio:device0/'

chan_file = 'in_voltage%i_raw'

convert(*bit_value*)

get(*ch=0*)

max_voltage = 0

volt_per_point = 0

Module contents

Objects Hardware Module This module collects objects and classes with hardware specific properties.

bacnet.object.hardware.**discover_hardware_objects**()
This function registers all defined objects in all hardware modules.

Returns object dictionary

Submodules

bacnet.object.define module

Object Define Module

This module contains user specific definitions of BACpypes Object and a list to collect all builtin Objects needed.

bacnet.object.define.**get_initial_object_list**()
This function creates the object list.

Returns object list

bacnet.object.primitivedata module

Primitive Data Module

This module contains improved bacpypes primitive data classes.

class bacnet.object.primitivedata.**COVSubscription**(**args*, ***kwargs*)
Bases: bacpypes.basetypes.COVSubscription

This class has specific updates.

sequenceElements = [<bacpypes.constructeddata.Element instance at 0x10586f0e0>, <bacpypes.constructeddata.Elen

class bacnet.object.primitivedata.**ObjectPropertyReference**(**args*, ***kwargs*)
Bases: bacpypes.constructeddata.Sequence

This class has an extension to support instances without property identifier.

sequenceElements = [<bacpypes.constructeddata.Element instance at 0x10586f1b8>, <bacpypes.constructeddata.Elen

class bacnet.object.primitivedata.**Remaining**(*time*)
Bases: bacpypes.primitivedata.CharacterString

This class is an extended version of Unsigned.

decode(*tag*)
This function decodes the value from network transmission.

Parameters **tag** – Tag

Returns None

encode(*tag*)
This function encodes the value for network transmission.

Parameters **tag** – Tag

Returns None

remaining_time
This function calculates the remaining time till defined datetime

Returns remaining time in seconds

class bacnet.object.primitivedata.**RingDict**(*size*, **args*, ***kwargs*)
Bases: collections.OrderedDict

This class is a dictionary ring buffer without overriding protection.

size
This function returns the size of the ring buffer.

Returns buffer size

class bacnet.object.primitivedata.**RingList**(*size*, *seq=()*)
Bases: list

This class is a ring buffer without overriding protection.

append(*data*)
This function adds data to ring buffer.

Parameters **data** – data to be added

Returns None

first_value()
This function returns the value last appended.

Returns last value

last_value()
This function returns the value last appended.

Returns last value

size
This function returns the size of the ring buffer.

Returns buffer size

class bacnet.object.primitivedata.**Unsigned**(**args*)
Bases: bacpypes.primitivedata.Unsigned, long

This class has an integer extension to support builtin functions like div float.

bacnet.object.properties module

Object Properties Module

This module contains specific properties for the usage within objects. Especially it provides updated property COV functionality for bacpypes objects.

class bacnet.object.properties.**ActiveCovSubscriptionsProperty**(**args*, ***kwargs*)
Bases: *bacnet.object.properties.ReadableProperty*

This class is an extension of the ReadableProperty class for active cov subscriptions.

ReadProperty(*obj*, *arrayIndex=None*, *dictionary=False*)
This function handles reading.

Parameters

- **obj** – property
- **arrayIndex** – index

Returns property value

WriteProperty(*obj*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function handles writing.

Parameters

- **obj** – property
- **value** – property value
- **arrayIndex** – index
- **priority** – priority
- **direct** – direct

Returns property value

class bacnet.object.properties.**CurrentDateProperty**(*identifier*)
Bases: *bacnet.object.properties.OptionalProperty*

This class is an extension of the OptionalProperty class for current date.

ReadProperty(*obj*, *arrayIndex=None*)
This function handles reading.

Parameters

- **obj** – property
- **arrayIndex** – index

Returns property value

WriteProperty(*obj*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function handles writing.

Parameters

- **obj** – property
- **value** – property value
- **arrayIndex** – index
- **priority** – priority
- **direct** – direct

Returns property value

class bacnet.object.properties.**CurrentTimeProperty**(*identifier*)
Bases: *bacnet.object.properties.OptionalProperty*

This class is an extension of the OptionalProperty class for current time.

ReadProperty(*obj*, *arrayIndex=None*)
This function handles reading.

Parameters

- **obj** – property
- **arrayIndex** – index

Returns property value

WriteProperty(*obj*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function handles writing.

Parameters

- **obj** – property
- **value** – property value

- **arrayIndex** – index
- **priority** – priority
- **direct** – direct

Returns property value

class bacnet.object.properties.**HardwareAccessObject**(*hw_object*, *write=False*, *hysteresis=False*, *buffer_size=0*)

Bases: object

This class is a wrapper for terra hardware objects.

get(*index=None*)
This function is a wrapper for getting the actual hardware value

Parameters **index** – array index

Returns value

hardware = None

hysteresis(*value*)
This function applies a user specific hysteresis to the value.

Parameters **value** – hardware value

Returns updated value

set(*value*, *data_type*, *index=None*)
This function converts the

Parameters

- **value** – value
- **data_type** – data type
- **index** – array index

Returns value

writable = False

class bacnet.object.properties.**ObjectIdentifierProperty**(*identifier*, *datatype*, *default=None*, *optional=False*, *mutable=False*, *cov_support=False*)

Bases: *bacnet.object.properties.ReadableProperty*

This class is an extension of the ReadableProperty class for object identifiers.

WriteProperty(*obj*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function handles writing.

Parameters

- **obj** – property
- **value** – property value
- **arrayIndex** – index
- **priority** – priority
- **direct** – direct

Returns property value

class bacnet.object.properties.**OptionalProperty**(*identifier*, *datatype*, *default=None*, *optional=True*, *mutable=False*, *cov_support=False*)
Bases: *bacnet.object.properties.StandardProperty*

This class is an extension of the bacpypes OptionalProperty class to support COV notifications.

class bacnet.object.properties.**Property**(*identifier*, *datatype*, ***kwargs*)
Bases: object

This class is an extension of the bacpypes Property class to support COV notifications.

ReadProperty(*obj*, *arrayIndex=None*)
This function handles reading.

Parameters

- **obj** – property
- **arrayIndex** – index

Returns property value

WriteProperty(*obj*, *value*, *arrayIndex=None*, *priority=None*, *direct=False*)
This function handles writing.

Parameters

- **obj** – property
- **value** – property value
- **arrayIndex** – index
- **priority** – priority
- **direct** – direct

Returns property value

class bacnet.object.properties.**ReadableProperty**(*identifier*, *datatype*, *default=None*, *optional=False*, *mutable=False*, *cov_support=False*)
Bases: *bacnet.object.properties.StandardProperty*

This class is an extension of the bacpypes ReadableProperty class to support COV notifications.

class bacnet.object.properties.**StandardProperty**(**args*, ***kwargs*)
Bases: *bacnet.object.properties.Property*

This class is an extension of the bacpypes StandardProperty class to support COV notifications.

class bacnet.object.properties.**WritableProperty**(*identifier*, *datatype*, *default=None*, *optional=False*, *mutable=True*, *cov_support=False*)
Bases: *bacnet.object.properties.StandardProperty*

This class is an extension of the bacpypes WritableProperty class to support COV notifications.

Module contents

Objects Module

This module contains just a single function to collect all user specific bacpypes Objects.

bacnet.object.**get_object_list**()
This function collects all defined user specific and relevant builtin bacpypes Objects.

Parameters **examples** – enable example objects

Returns object_list

1.1.6 bacnet.sandbox package

Subpackages

Submodules

bacnet.sandbox.interface module

Sandbox Interface Module

This module contains the Transformation Implementation for the Sandbox interface. It is simply copied from the PyPy code and thereby not commented.

bacnet.sandbox.interface.**read_message**(*timeout=None*)

bacnet.sandbox.interface.**receive**()
This function is waiting for arriving messages via an event.

Returns message

bacnet.sandbox.interface.**transmit**(*line*)
This function is transmitting outgoing messages.

Returns None

bacnet.sandbox.interface.**write_message**(**x*)

bacnet.sandbox.process module

Module contents

Sandbox Module

This module contains handlers for the pypy sandbox environment.

bacnet.sandbox.**get_platform_name**()
This function returns the platform name.

Returns platform name

bacnet.sandbox.**get_sandbox_process**(*executable*)
This function returns a sandbox process object.

Parameters **executable** – relative path to untrusted executable

Returns sandbox process

bacnet.sandbox.**sandbox_interact**(*sandbox_process*)
This function interacts with the sandbox process.

Parameters **sandbox_process** – sandbox process object

Returns None

bacnet.sandbox.**sandbox_support**(*platform_name='darwin-x86_64'*)
This function checks if compiled sandbox version of pypy exists for this system.

Returns sandbox is supported on this system

1.1.7 bacnet.system package

Subpackages

bacnet.system.config package

Submodules

bacnet.system.config.helper module

Handler Helper Module This module provides access to local interface addresses.

bacnet.system.config.helper.**get_local_ip**(*interface*)
This function retrieves an IP address from the defined interface.

Parameters **interface** – network interface for IP address retrieval

Returns ip_address

bacnet.system.config.parser module

Config Parser Module This module provides an extended version of the BCApypes argument parser.

class bacnet.system.config.parser.**FullArgumentParser**(***kwargs*)
Bases: argparse.ArgumentParser

This class combines BACpypes builtin config and parameter parser with customized parameters.

arg_object = None

parse_args(**args*, ***kwargs*)
This function wraps the parse_args of ConfigArgumentParser to intercept version requests.

Returns self

update()
This function updates the argument object.

Returns parsed parameters

Module contents

BACnet Config Module This module provides configuration functionality.

class bacnet.system.config.**ConfigManager**(***kwargs*)
Bases: object

This class handles the configuration requests by the user.

data = {}

filename = 'BACnet.ini'

get(*key*)
This function retrieves specified value provided by the user.

Parameters **key** – value to retrieve

Returns None

get_input(*key*, *prompt*)
This function handles user input.

Parameters

- **key** – default key
- **prompt** – prompt text

Returns user response

start()
This function initiates the user interface.

Returns None

store()
This function stores the retrieved data.

Returns None

valid(*key*, *response*)
This function checks, if response is valid

Parameters

- **key** – value to retrieve
- **response** – retrieved value

Returns

verify_address(*key='address'*, *response=''*, *interface=True*)
This function retrieves an ip address provided by the user.

Parameters **response** – user input

Returns verified response

verify_bbmd(*response*)
This function verifies an ip address provided by the user.

Parameters **response** – user input

Returns verified response

Submodules

bacnet.system.managing module

BACnet Managing Module

This module provides managing tools for multiprocessing.

bacnet.system.managing.**client_manager**(*address=None*, *authkey=None*)
This function creates a client manager

Returns manager

bacnet.system.managing.**server_manager**(*log_queue*, *ip_address*, *port*, *console=False*, *webgui=False*)
This function creates a server manager.

Returns manager, server

bacnet.system.vendors module

Vendors

This module provides an updater for BACnet vendor IDs.

class bacnet.system.vendors.**TableParser**
Bases: htmllib.HTMLParser

This class implements a table parser for the BACnet vendor list.

end_tr()
This function handles the end of a column.

Returns

handle_data(*data*)
This function handles tag data.

Parameters **data** – tag data

Returns None

handle_starttag(*tag*, *method*, *attrs*)
This function is called on every start tag.

Parameters

- **tag** – full tag
- **method** – tag function
- **attrs** – tag attributes

Returns None

handle_td(*data*)
This function handles a new column.

Parameters **data** – tag content

Returns None

start_td(*attrs*)
This function handles a new column.

Parameters **attrs** – tag attributes

Returns None

bacnet.system.vendors.**get_vendors**()
This function requests the current vendor list and returns a parsed dictionary

Returns vendor dictionary

bacnet.system.version module

BACnet System Version Module

This module handles version printing and normalizing according to PEP 386.

bacnet.system.version.**create_version**(*version*)
This function is a verlib wrapper to create necessary version and git information.

Parameters **version** – get current version tuple

Returns full_version, git_info

bacnet.system.version.**get_version**()
This function returns current version string and exits program.

Returns None

Module contents

BACnet System Module

This module contains the bacnet system class.

class bacnet.system.**BACnetSystem**(*stdin=<open file '<stdin>', mode 'r'>, stdout=<open file '<stdout>', mode 'w'>, stderr=<open file '<stderr>', mode 'w'>*)
Bases: object

This class handles the BACnet system.

application = None

console = None

console_interrupt(**args*)
This function is the catch for interrupts.

Returns None

do_buggers(*config_args*)
This function lists all buggers.

Parameters **config_args** – configuration arguments

Returns None

do_config(*config_args*)
This function sets configuration parameters.

Parameters **config_args** – configuration arguments

Returns None

do_hardware(*config_args*)
This function lists all creatable hardware objects.

Parameters **config_args** – configuration arguments

Returns None

do_set_tag(*config_args*)
This function initiates the BACnet system start up as shell.

Parameters **config_args** – configuration arguments

Returns None

do_shell(*config_args*)
This function initiates the BACnet system start up as shell.

Parameters **config_args** – configuration arguments

Returns None

do_start(*config_args*, *console=False*, *webgui=False*)
This function initiates the BACnet system start up as reactive application.

Parameters **config_args** – configuration arguments

Returns None

do_webgui(*config_args*)
This function initiates the BACnet system start up as web gui.

Parameters **config_args** – configuration arguments

Returns None

exit(*code=0*)
This function exits the system.

Parameters **code** – exit code

Returns None

manager = None

shutdown(**args*)
This function shuts down the process.

Returns None

stderr = None

stdin = None

stdout = None

1.1.8 bacnet.webgui package

Subpackages

bacnet.webgui.caching package

Subpackages

bacnet.webgui.caching.base package

Submodules

bacnet.webgui.caching.base.filter module

bacnet.webgui.caching.base.helpers module

bacnet.webgui.caching.base.options module

bacnet.webgui.caching.base.relations module

Module contents

Submodules

bacnet.webgui.caching.cache module

Module contents

bacnet.webgui.control package

Subpackages

bacnet.webgui.control.templatetags package

Submodules

bacnet.webgui.control.templatetags.menu module

Menu Templatetags Module This module contains important descriptions for the menu.

bacnet.webgui.control.templatetags.menu.**url_attr**(*context*, *label*, *attribute*, *exact=False*, ***kwargs*)

This function marks links depending on the request's path.

Module contents

Control Templatetags Module This module is a collection of all tags used within the app's templates.

bacnet.webgui.control.views package

Submodules

bacnet.webgui.control.views.access module

bacnet.webgui.control.views.devices module

Module contents

Views Module This module contains all descriptions of web access interfaces.

bacnet.webgui.control.views.**breadcrumbs**(*path*)
This function provides the breadcrumbs for a path.

Parameters **path** – path

Returns list of breadcrumbs

bacnet.webgui.control.views.**control_views_render**(*request*, *template_file*, *arg_dict={}*, *local=True*)
This function provides general rendering for all templates.

Parameters

- **request** – Django request
- **template_file** – template filename
- **arg_dict** – arguments
- **local** – is local template

Returns HTTP response

bacnet.webgui.control.views.**get_crumb_name**(*link*)
This function provides the breadcrumb name for a link

Parameters **link** – link

Returns breadcrumb name

bacnet.webgui.control.views.**index**(*request*)
This function provides the index page of the project.

Parameters **request** – Django request

Returns HTTP response

Submodules

bacnet.webgui.control.models module

bacnet.webgui.control.urls module

Control URL Module This module contains the description of all app based pages on the web interface.

Module contents

Control Module This module contains all necessary descriptions for the web interface and internal structure.

bacnet.webgui.webgui package

Subpackages

bacnet.webgui.webgui.settings package

Submodules

bacnet.webgui.webgui.settings.apps module

App Settings Module This module contains app specific definitions for Django settings.

bacnet.webgui.webgui.settings.private module

Private Settings Module This module contains private definitions for Django settings.

Module contents Django settings for webgui project.

For more information on this file, see https://docs.djangoproject.com/en/1.7/topics/settings/

For the full list of settings and their values, see https://docs.djangoproject.com/en/1.7/ref/settings/

Submodules

bacnet.webgui.webgui.urls module

bacnet.webgui.webgui.wsgi module WSGI config for webgui project.

It exposes the WSGI callable as a module-level variable named `application`.

For more information on this file, see https://docs.djangoproject.com/en/1.7/howto/deployment/wsgi/

Module contents

Submodules

bacnet.webgui.bacnet_access module

BACnet Access Module

This module provides access to the inter-process communication of the system.

`bacnet.webgui.bacnet_access.`**`api_create`**(*line*)

This function wraps the API method create.

Parameters **`line`** – command

Returns APDU object

bacnet.webgui.bacnet_access.**api_receive**()
This function wraps the API methods receive and parse.

Returns parsed APDU object

bacnet.webgui.bacnet_access.**api_transmit**(*line=None*, *request=None*)
This function wraps the API methods create and send.

Parameters

- **line** – command
- **request** – APDU object

Returns sending successful

bacnet.webgui.manage module

Manage Module

This module manages the django environment and process start up.

Module contents

BACnet WebGUI Module

This module contains a django project for BACnet interaction.

class bacnet.webgui.**DjangoProcess**(*address*, *port*)
Bases: multiprocessing.process.Process

This class handles the django process.

receiver()
This function provides the environment for the receiver thread.

Returns None

run()
This function is executed as a separate process.

shutdown(**args*)
This function shuts down the process.

Returns None

bacnet.webgui.**create_webgui**(*address*, *port*)
This function creates a django subprocess.

Returns webgui process

1.2 Submodules

1.3 bacnet.settings module

1.3.1 BACnet Settings Module

This module contains general settings for the BACnet system.

1.4 Module contents

1.4.1 BACnet Framework

This module contains wrappers for BACpypes to ensure seamless integration of BACnet functionality to user specific use cases.

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

b

A

B

C

D

E

F

G

H

P

R

S

T

U

V

W