
secsgem Documentation

Release 0.1.0

Benjamin Parzella

Jun 06, 2021

Contents

1	Namespaces	3
2	Thanks	5
3	Table of contents	7
3.1	Installation	7
3.2	First Steps	7
3.2.1	Callbacks and events	7
3.2.1.1	Callbacks	7
3.2.1.2	Events	9
3.2.2	Implementing a GEM equipment	10
3.2.2.1	Using your own name	10
3.2.2.2	Adding status variables	11
3.2.2.3	Adding equipment constants	12
3.2.2.4	Adding collection events	13
3.2.2.5	Adding alarms	13
3.2.2.6	Adding remote commands	14
3.2.3	Custom streams and functions	14
3.2.3.1	Custom data item	14
3.2.3.2	Custom stream function	15
3.2.3.3	Integrate a stream function	15
3.2.4	Testing	15
3.3	GEM	16
3.3.1	Handlers	16
3.3.1.1	Events	17
3.3.2	GEM Compliance	18
3.3.2.1	State Models	18
3.3.2.2	Equipment Processing States	18
3.3.2.3	Documentation	18
3.3.2.4	Control (Operator Initiated)	18
3.3.2.5	Dynamic Event Report Configuration	19
3.3.2.6	Trace Data Collection	19
3.3.2.7	Alarm Management	19
3.3.2.8	Remote Control	19
3.3.2.9	Equipment Constants	19
3.3.2.10	Process Recipe Management	19
3.3.2.11	Material Movement	19

	3.3.2.12	Equipment Terminal Services	19
	3.3.2.13	Clock	19
	3.3.2.14	Limits Monitoring	19
	3.3.2.15	Spooling	19
3.4	SECS	20
	3.4.1	Variables	20
	3.4.1.1	Type arrays	20
	3.4.1.2	Getting data	21
	3.4.1.3	Setting data	21
	3.4.1.4	En-/Decoding	22
	3.4.1.5	SecsVarArray	22
	3.4.1.6	SecsVarList	22
	3.4.1.7	SecsVarDynamic	23
	3.4.2	Functions	23
	3.4.3	Handler	25
3.5	HSMS	25
	3.5.1	Packets	25
	3.5.1.1	Select Request	26
	3.5.1.2	Select Response	26
	3.5.1.3	Deselect Request	26
	3.5.1.4	Deselect Response	26
	3.5.1.5	Linktest Request	27
	3.5.1.6	Linktest Response	27
	3.5.1.7	Reject Request	27
	3.5.1.8	Separate Request	27
	3.5.1.9	Data Message	27
	3.5.2	Connections	27
	3.5.2.1	Delegates	28
	3.5.2.2	Active connection	28
	3.5.2.3	Passive connection	29
	3.5.2.4	Multi-passive connection	29
	3.5.2.5	Connection manager	29
	3.5.3	Handler	30
	3.5.3.1	Events	30
3.6	Class reference	31
	3.6.1	HSMS	31
	3.6.1.1	Packets	31
	3.6.1.2	Connections	35
	3.6.1.3	Handler	38
	3.6.1.4	ConnectionManager	39
	3.6.2	SECS	40
	3.6.2.1	Variables	40
	3.6.2.2	Data Items	68
	3.6.2.3	FunctionBase	122
	3.6.2.4	Functions	122
	3.6.2.5	Handler	188
	3.6.3	GEM	193
	3.6.3.1	Handler	193
	3.6.3.2	HostHandler	199
	3.6.3.3	EquipmentHandler	206
	3.6.4	Common functionality	213

secsgem is a python package to communicate with a host or equipment system in the semiconductor industry.

The use cases range from writing tests for implementations or features, simulations in development environments to complete host/equipment implementations. Some parts of the package can be used individually, for example HSMS can be used without SECS-II, or the streams and functions can be used with a different networking stack.

Currently there is no support for communication over serial port (SECS-I, SEMI E04). Only ethernet (HSMS, SEMI E37) is available.

HSMS, SECS and GEM are standards from [SEMI](#).

CHAPTER 1

Namespaces

All classes can be accessed with their full module name or directly from the secsgem module.

```
>>> secsgem.format_hex("Hallo")  
'48:61:6c:6c:6f'
```

```
>>> secsgem.common.format_hex("Hello")  
'48:65:6c:6c:6f'
```


CHAPTER 2

Thanks

- [Carl Wolff](#) for his sample TCP socket implementation
- [Darius Sullivan](#) for his [gist](#) on how to use the streams and functions with twisted
- [Massimo Vanetti](#) for his help on the equipment implementation

3.1 Installation

Official releases are available via [Pypi](#) repository. The easiest way to install these is using a package manager like pip:

```
$ pip install secsgem
```

In order to use the current development code, which might be instable at times, use the git repository directly:

```
$ pip install git+git://github.com/bparzella/secsgem
```

3.2 First Steps

secsgem can be used in different ways. You can use it to create a GEM implementation for either equipment and host side. But also implementations on the SECS-II and HSMS levels are possible. Another way to use it is for testing your host or equipment implementation using python unit tests.

3.2.1 Callbacks and events

Callbacks are used to handle requests from the remote system and return a user defined result (eg an alarm has been received and a response is required). *Events* can notify the implementation about something that occurred (eg the hsms connection was selected). Events don't return any result to the remote and are executed in the background

3.2.1.1 Callbacks

Callbacks are used to request information from a specific implementation. They can be used to process the passed information and present a result to the peer. Only one function can be registered for one callback.

The process will wait for the callback to return the result of the calculation. Because of that the callback should run as performant as possible.

There are three ways to define the callback functions, by creating them in the *inherited handler*, by setting a *target object* and by *registering callbacks*. Registered callbacks supersede target and overridden functions.

Inherited handler

When working with a inherited class, callbacks can be implemented by creating callback members with a specific name:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

    def _on_alarm_received(self, handler, ALID, ALCD, ALTX):
        return ACKC5.ACCEPTED
```

In this example when an alarm was received (callback name is alarm_received) the `_on_alarm_received` method will be called. The result (in this case `ACKC5.ACCEPTED`) will be passed to the host sending the alarm.

A generic representation of the function would be:

```
def _on_<callback_name>(self, handler, <parameters>):
    return <result>
```

Callbacks for streams/functions can also be overridden this way by following a specific naming:

```
def _on_s05f01(self, handler, packet):
    return self.stream_function(5, 2)(ACKC5.ACCEPTED)
```

Note that the stream and function numbers are formatted to have a leading zero if they are only one character long. In this case the reply stream/function must be returned.

Target object

These methods don't need to be implemented on the handler itself. Another object can also be registered using the callbacks member names of the handler. The `_on_<callback_name>` methods are then searched in that object:

```
class TestClass(object):
    def _on_alarm_received(self, handler, ALID, ALCD, ALTX):
        return ACKC5.ACCEPTED

t = TestClass()

handler.callbacks.target = t
```

Registering callbacks

Callbacks can also be registered from outside a class:

```
def f_alarm_received(handler, ALID, ALCD, ALTX):
    return ACKC5.ACCEPTED

handler.callbacks.alarm_received = f_alarm_received
```

To unregister simply clear the member:

```
handler.callbacks.alarm_received = None
```

Available callbacks

3.2.1.2 Events

Events will notify the implementation that things happened. They are called asynchronously, the result will be ignored.

There are three ways to define events, by creating them in the *inherited handler*, by setting *target objects* in the handlers events property and by *registering events*.

Inherited handler

When working with a inherited class, events can be implemented by creating members with a specific name:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

    def _on_event_hsms_selected(self, connection):
        pass
```

In this example when the hsms connection state changes to selected the `_on_event_hsms_selected` method will be called.

A generic representation of the function would be:

```
def _on_event_<event_name>(self, <parameters>):
    pass
```

To catch all events, the `_on_event` method can be overridden:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

    def _on_event(self, *args):
        pass
```

Target objects

These methods don't need to be implemented on the handler itself. Other objects can also be registered using the event member names of the handler. The `_on_event_<event_name>` and `_on_event` methods are then searched in that object:

```
class TestClass(object):
    def _on_event_hsms_selected(self, connection):
        pass

t = TestClass()

handler.events.targets += t
```

The event handler can work with more than one target objects.

Registering events

Events can also be registered from outside a class:

```
def f_hsms_selected(connection):
    pass

handler.events.hsms_selected += f_hsms_selected
```

To unregister simply remove the member:

```
handler.events.hsms_selected -= f_hsms_selected
```

Available events

3.2.2 Implementing a GEM equipment

This package can be used to create a GEM equipment implementation. This is done by subclassing the *secsgem.gem.equipmenthandler.GemEquipmentHandler* class:

```
import secsgem
import code

class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

h = SampleEquipment("127.0.0.1", 5000, False, 0, "sampleequipment")
h.enable()

code.interact("equipment object is available as variable 'h', press ctrl-d to stop",
↳ local=locals())

h.disable()
```

3.2.2.1 Using your own name

To use your own modelname and version for S1F14 reply you can override the *secsgem.gem.handler.GemHandler.MDLN* and *secsgem.gem.handler.GemHandler.SOFTREV* members of the GemHandler:


```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.MDLN = "gemequip"
        self.SOFTREV = "1.0.0"
```

3.2.2.2 Adding status variables

A status variable can be added by inserting an instance of the `secsgem.gem.equipmenthandler.StatusVariable` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.status_variables` dictionary:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.status_variables.update({
            10: secsgem.StatusVariable(10, "sample1, numeric SVID, SecsVarU4", "meters
↳ ", secsgem.SecsVarU4, False),
            "SV2": secsgem.StatusVariable("SV2", "sample2, text SVID, SecsVarString",
↳ "chars", secsgem.SecsVarString, False),
        })

        self.status_variables[10].value = 123
        self.status_variables["SV2"].value = "sample sv"
```

Alternatively the values can be acquired using a callback by setting the `use_callback` parameter of the constructor to `True`:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.sv1 = 123
        self.sv2 = "sample sv"

        self.status_variables.update({
            10: secsgem.StatusVariable(10, "sample1, numeric SVID, SecsVarU4", "meters
↳ ", secsgem.SecsVarU4, True),
            "SV2": secsgem.StatusVariable("SV2", "sample2, text SVID, SecsVarString",
↳ "chars", secsgem.SecsVarString, True),
        })

    def on_sv_value_request(self, svid, sv):
        if sv.svid == 10:
            return sv.value_type(value=self.sv1)
        elif sv.svid == "SV2":
            return sv.value_type(value=self.sv2)
```

(continues on next page)

(continued from previous page)

```
return []
```

3.2.2.3 Adding equipment constants

An equipment constant can be added by inserting an instance of the `secsgem.gem.equipmenthandler.EquipmentConstant` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.status_variables` dictionary:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.equipment_constants.update({
            20: secsgem.EquipmentConstant(20, "sample1, numeric ECID, SecsVarU4", 0,
↳ 500, 50, "degrees", secsgem.SecsVarU4, False),
            "EC2": secsgem.EquipmentConstant("EC2", "sample2, text ECID, SecsVarString
↳ ", "", "", "chars", secsgem.SecsVarString, False),
        })

        self.status_variables[20].value = 321
        self.status_variables["EC2"].value = "sample ec"
```

Alternatively the values can be acquired and updated using callbacks by setting the `use_callback` parameter of the constructor to `True`:

```
class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.ec1 = 321
        self.ec2 = "sample ec"

        self.equipment_constants.update({
            20: secsgem.EquipmentConstant(20, "sample1, numeric ECID, SecsVarU4", 0,
↳ 500, 50, "degrees", secsgem.SecsVarU4, True),
            "EC2": secsgem.EquipmentConstant("EC2", "sample2, text ECID, SecsVarString
↳ ", "", "", "chars", secsgem.SecsVarString, True),
        })

    def on_ec_value_request(self, ecid, ec):
        if ec.ecid == 20:
            return ec.value_type(value=self.ec1)
        elif ec.ecid == "EC2":
            return ec.value_type(value=self.ec2)

        return []

    def on_ec_value_update(self, ecid, ec, value):
        if ec.ecid == 20:
            self.ec1 = value
```

(continues on next page)

(continued from previous page)

```

elif ec.ecid == "EC2":
    self.ec2 = value

```

3.2.2.4 Adding collection events

A collection event can be added by inserting an instance of the `secsgem.gem.equipmenthandler.CollectionEvent` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.collection_events` dictionary. Data values can be added by inserting an instance of the `secsgem.gem.equipmenthandler.DataValue` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.data_values` dictionary. The data values for a collection event can be passed while creating the `secsgem.gem.equipmenthandler.CollectionEvent` instance:

```

class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.dv1 = 31337

        self.data_values.update({
            30: secsgem.DataValue(30, "sample1, numeric DV, SecsVarU4", secsgem.
↳ SecsVarU4, True),
        })

        self.collection_events.update({
            50: secsgem.CollectionEvent(50, "test collection event", [30]),
        })

    def on_dv_value_request(self, dvid, dv):
        if dv.dvid == 30:
            return dv.value_type(value=self.dv1)

        return []

    def trigger_sample_collection_event():
        self.trigger_collection_events([50])

```

3.2.2.5 Adding alarms

An alarm can be added by inserting an instance of the `secsgem.gem.equipmenthandler.Alarm` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.alarms` dictionary. The collection events for the alarm must be provided when adding the alarm. For an example see the section above:

```

class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↳ handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↳ name, custom_connection_handler)

        self.collection_events.update({
            100025: secsgem.CollectionEvent(100025, "test collection event alarm set",
↳ []),

```

(continues on next page)

(continued from previous page)

```

        200025: secsgem.CollectionEvent(200025, "test collection event alarm clear
↪", []),
    })

    self.alarms.update({
        25: secsgem.Alarm(25, "test alarm", "test text", secsgem.ALCD.PERSONAL_
↪SAFETY | secsgem.ALCD.EQUIPMENT_SAFETY, 100025, 200025),
    })

    def set_sample_alarm():
        self.set_alarm(25)

    def clear_sample_alarm():
        self.clear_alarm(25)

```

3.2.2.6 Adding remote commands

A remote command can be added by inserting an instance of the `secsgem.gem.equipmenthandler.RemoteCommand` class to the `secsgem.gem.equipmenthandler.GemEquipmentHandler.remote_commands` dictionary. The collection event and parameters for the remote command must be provided when adding it. For an example see the section above:

```

class SampleEquipment(secsgem.GemEquipmentHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
↪handler=None):
        secsgem.GemEquipmentHandler.__init__(self, address, port, active, session_id,
↪name, custom_connection_handler)

        self.collection_events.update({
            5001: secsgem.CollectionEvent(5001, "TEST_RCMD complete", []),
        })
        self.remote_commands.update({
            "TEST_RCMD": secsgem.RemoteCommand("TEST_RCMD", "test rcmd", ["TEST_
↪PARAMETER"], 5001),
        })

    def on_rcmd_TEST_RCMD(self, TEST_PARAMETER):
        print "remote command TEST_RCMD received"

```

3.2.3 Custom streams and functions

3.2.3.1 Custom data item

A new data item is created by overriding the `DataItemBase` class:

```

class UNITS_New(DataItemBase):
    __type__ = SecsVarDynamic
    __allowedtypes__ = [SecsVarArray, SecsVarBoolean, SecsVarU1, SecsVarU2, SecsVarU4,
↪ SecsVarU8, SecsVarI1, SecsVarI2, SecsVarI4, SecsVarI8, \
        SecsVarF4, SecsVarF8, SecsVarString, SecsVarBinary]

```

In this case the UNITS field allows all types instead only a string.

3.2.3.2 Custom stream function

To integrate this new data item in a stream function then you need to inherit `secsgem.secs.functionbase.SecsStreamFunction`

```
class SecsS01F12_New(secsgem.SecsStreamFunction):
    _stream = 1
    _function = 12

    _dataFormat = [
        [
            SVID,
            SVNAME,
            UNITS_New
        ]
    ]

    _toHost = True
    _toEquipment = False

    _hasReply = False
    _isReplyRequired = False

    _isMultiBlock = True
```

3.2.3.3 Integrate a stream function

Now we want to integrate this stream/function into the `secsgem.gem.handler.GemHandler`. You create a new class inherited from it and update the function list of that class:

```
class NewHandler(secsgem.GemHostHandler):
    def __init__(self, address, port, active, session_id, name, custom_connection_
    ↪ handler=None):
        secsgem.GemHostHandler.__init__(self, address, port, active, session_id, name,
    ↪ custom_connection_handler)

        self.secsStreamsFunctions[1].update({
            12: SecsS01F12_New,
        })
```

You can also add new methods and properties to the class if required.

This new class can be used with the `secsgem.hsms.connectionmanager.HsmsConnectionManager` (see parameter `connection_handler` of `secsgem.hsms.connectionmanager.HsmsConnectionManager.add_peer()`)

3.2.4 Testing

secsgem can be used to make unit tests on your implementation of the SEMI standard.

Example:

```
import unittest

import secsgem
```

(continues on next page)

(continued from previous page)

```
class TestExampleSecsGem(unittest.TestCase):
    def setUp(self):
        self.connection = secsgem.GemHostHandler("10.211.55.33", 5000, False, 0, "test
→")

        self.connection.enable()
        self.connection.waitFor_communicating()

    def tearDown(self):
        self.connection.disable()

    def testLinktest(self):
        result_packet = self.connection.send_linktest_req()

        self.assertEqual(result_packet.header.sType, 6)
        self.assertEqual(result_packet.header.sessionID, 65535)
```

See file samples/testExample.py

3.3 GEM

SEMI 30

GEM defines certain behaviors of the equipment and how to use the SECS messages for that purpose.

3.3.1 Handlers

secsgem.gem.handler.GemHandler inherits the functionality from *secsgem.secs.handler.SecsHandler* (see *Handler*).

To distinguish between host and equipment process there are two specialized types of *secsgem.gem.handler.GemHandler*: *secsgem.gem.hosthandler.GemHostHandler* and *secsgem.gem.equipmenthandler.GemEquipmentHandler*. Use *GemHostHandler* if you want to implement a host system, *GemEquipmentHandler* for a equipment system.

It automatically handles the whole setup and teardown of the link. Incoming collection events and terminal messages are automatically accepted and propagated by events. The setup of collection event reports is also simplified. It has functionality to send remote commands and handling process programs.

The handler also implements a maintains a communication state, which is defined in the standard.

```
>>> client = secsgem.GemHostHandler("10.211.55.33", 5000, False, 0, "test")
>>>
>>> client.enable()
>>> client.waitFor_communicating()
True
>>> client.get_process_program_list()
['test1', 'test2']
>>> client.request_process_program('test1')
This is process program test1
>>> client.disable()
```

Waiting for the communicating state can also be done asynchronous

```

>>> def on_communicating(event, data):
...     print "Communicating"
...
>>> client = secsgem.GemHostHandler("10.211.55.33", 5000, False, 0, "test")
>>> client.events.handler_communicating += on_communicating
>>>
>>> client.enable()
Communicating
>>> client.get_process_program_list()
['test1', 'test2']
>>> client.request_process_program('test1')
This is process program test1
>>> client.disable()

```

Also streams/functions can be sent and received with the handler:

```

>>> f = secsgem.SecsS01F01()
>>> client.send_and_waitfor_response(f)
secsgem.hsms.packets.HsmsPacket({'header': secsgem.hsms.packets.HsmsHeader({'function
↪': 2, 'stream': 1, 'pType': 0, 'system': 14, 'sessionID': 0, 'requireResponse': ↪
↪False, 'sType': 0}), 'data': '\x01\x02A\x06EQUIPMA\x06SV n/a'})

```

3.3.1.1 Events

GemHandler defines a few new events, that can be received with the help of `secsgem.common.EventHandler`:

Event name	Description
handler_communicating	Connection is setup
collection_event_received	Collection event was received
terminal_received	Terminal message was received

For an example on how to use these events see the code fragment in *Handler*.

3.3.2 GEM Compliance

GEM COMPLIANCE STATEMENT				
Fundamental GEM Requirements		Implemented	GEM Compliant	
State Models		Yes ✓	No	No
Equipment Processing States		No	No	
Host-Initiated S1,F13/F14 Scenario		Yes ✓	Yes ✓	
Event Notification		Yes ✓	Yes ✓	
On-Line Identification		Yes ✓	Yes ✓	
Error Messages		Yes ✓	Yes ✓	
Documentation		Yes ✓	No	
Control (Operator Initiated)		Yes ✓	No	
Additional Capabilities		Implemented	GEM Compliant	
Establish Communications		Yes ✓	Yes ✓	
Dynamic Event Report Configuration		Yes ✓	No	
Variable Data Collection		Yes ✓	Yes ✓	
Trace Data Collection		No	No	
Status Data Collection		Yes ✓	Yes ✓	
Alarm Management		Yes ✓	No	
Remote Control		Yes ✓	Yes ✓	
Equipment Constants		Yes ✓	No	
Process Recipe Management		No	No	
Material Movement		No	No	
Equipment Terminal Services		Yes ✓	Yes ✓	
Clock		No	No	
Limits Monitoring		No	No	
Spooling		No	No	
Control (Host-Initiated)		Yes ✓	Yes ✓	

3.3.2.1 State Models

- While the communication and control state models are implemented, especially the control state model needs rework.

3.3.2.2 Equipment Processing States

- Not implemented yet.

3.3.2.3 Documentation

- The documentation isn't complete yet.

3.3.2.4 Control (Operator Initiated)

- Persistence for the ONLINE LOCAL/REMOTE is not yet implemented.
- The final UI (or hardware) needs the buttons required by this section.

3.3.2.5 Dynamic Event Report Configuration

- Persistence for report definitions, report-to-event links and enable status is not yet implemented.

3.3.2.6 Trace Data Collection

- Not implemented yet.

3.3.2.7 Alarm Management

- Persistence of en-/disable states and report definitions is not implemented yet.

3.3.2.8 Remote Control

- The START and STOP remote commands must be implemented to be GEM compliant. Currently only dummy functions are provided

3.3.2.9 Equipment Constants

- Persistence of the equipment constants is not implemented yet.
- Limiting changing equipment to “safe” states is not yet implemented?
- Equipment constant changed collection event is not yet implemented.

3.3.2.10 Process Recipe Management

- Not implemented yet.

3.3.2.11 Material Movement

- Not implemented yet.

3.3.2.12 Equipment Terminal Services

- The UI requirements can't be fulfilled by the library

3.3.2.13 Clock

- Not implemented yet.

3.3.2.14 Limits Monitoring

- Not implemented yet.

3.3.2.15 Spooling

- Not implemented yet.

3.4 SECS

SEMI E5

SECS-II defines the messages the data is transferred in between host and equipment over the HSMS protocol (and SECS-I serial). It specifies data types that contain the data and streams and functions that use these types for specific purposes.

3.4.1 Variables

SECS defines a few types to transmit data in.

Data Type	Class	Code
List	<code>secsgem.secs.variables.SecsVarArray</code>	L
List	<code>secsgem.secs.variables.SecsVarList</code>	L
Binary	<code>secsgem.secs.variables.SecsVarBinary</code>	B
Boolean	<code>secsgem.secs.variables.SecsVarBoolean</code>	TF
ASCII	<code>secsgem.secs.variables.SecsVarString</code>	A
8-Byte integer	<code>secsgem.secs.variables.SecsVarI8</code>	I8
1-Byte integer	<code>secsgem.secs.variables.SecsVarI1</code>	I1
2-Byte integer	<code>secsgem.secs.variables.SecsVarI2</code>	I2
4-Byte integer	<code>secsgem.secs.variables.SecsVarI4</code>	I4
8-Byte float	<code>secsgem.secs.variables.SecsVarF8</code>	F8
4-Byte float	<code>secsgem.secs.variables.SecsVarF4</code>	F8
8-Byte unsigned integer	<code>secsgem.secs.variables.SecsVarU8</code>	U8
1-Byte unsigned integer	<code>secsgem.secs.variables.SecsVarU1</code>	U1
2-Byte unsigned integer	<code>secsgem.secs.variables.SecsVarU2</code>	U2
4-Byte unsigned integer	<code>secsgem.secs.variables.SecsVarU4</code>	U4

Example:

```
>>> secsgem.SecsVarString("TESTString")
<A "TESTString">
>>> secsgem.SecsVarBoolean(True)
<BOOLEAN True >
>>> secsgem.SecsVarU4(1337)
<U4 1337 >
```

3.4.1.1 Type arrays

The numeric types can also be an array of that type:

```
>>> secsgem.SecsVarU1([1, 2, 3, 4])
<U1 1 2 3 4 >
>>> secsgem.SecsVarBoolean([True, False, False, True])
<BOOLEAN True False False True >
```

The length of this array can be fixed with the length parameter:

```
>>> secsgem.SecsVarU1([1, 2, 3], count=3)
<U1 1 2 3 >
>>> secsgem.SecsVarU1([1, 2, 3, 4], count=3)
```

(continues on next page)

(continued from previous page)

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/ext005207/Development/secsgem/secsgem/secs/variables.py", line 1439, in
↪__init__
    self.set(value)
  File "/home/ext005207/Development/secsgem/secsgem/secs/variables.py", line 1537, in
↪set
    raise ValueError("Value longer than {} chars".format(self.count))
ValueError: Value longer than 3 chars
```

```
>>> secsgem.SecsVarString("Hello", count=3).get()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/ext005207/Development/secsgem/secsgem/secs/variables.py", line 1220, in
↪__init__
    self.set(value)
  File "/home/ext005207/Development/secsgem/secsgem/secs/variables.py", line 1337, in
↪set
    raise ValueError("Value longer than {} chars ({} chars)".format(self.count,
↪len(value)))
ValueError: Value longer than 3 chars (5 chars)
```

3.4.1.2 Getting data

The data can be accessed with the `secsgem.secs.variables.SecsVarU1.get()` method, arrays can be accessed using the index operator:

```
>>> secsgem.SecsVarU1(1).get()
1
>>> secsgem.SecsVarU1([1, 2, 3], count=3).get()
[1, 2, 3]
>>> secsgem.SecsVarU1(1)[0]
1
>>> secsgem.SecsVarU1([1, 2, 3])[1]
2
```

3.4.1.3 Setting data

The data can be set with the `secsgem.secs.variables.SecsVarString.set()` method, arrays can be updated using the index operator:

```
>>> v=secsgem.SecsVarU1([1, 2, 3], count=3)
>>> v.set([3, 2, 1])
>>> v
<U1 3 2 1 >
>>> v[0] = 1
>>> v
<U1 1 2 1 >
```

3.4.1.4 En-/Decoding

The variable types can `secsgem.secs.variables.SecsVarArray.encode()` and `secsgem.secs.variables.SecsVarString.decode()` themselves to ASCII data transferrable with the HSMS protocol:

```
>>> v=secsgem.SecsVarString("Hello")
>>> d=v.encode()
>>> d
'A\x05Hello'
>>> secsgem.format_hex(d)
'41:05:48:65:6c:6c:6f'
>>> v.set("NewText")
>>> v
<A "NewText">
>>> v.decode(d)
7
>>> v
<A "Hello">
```

3.4.1.5 SecsVarArray

`secsgem.secs.variables.SecsVarArray` is a special type for a list of the same type. The items of the array can be accessed with the index operator.

```
>>> v=secsgem.SecsVarArray(secsgem.SecsVarU4)
>>> v.set([1, 2, 3])
>>> v
<L [3]
<U4 1 >
<U4 2 >
<U4 3 >
```

```
> >>> v.get([1, 2, 3]) >>> v[1] <U4 2 >
```

A new item can be appended to the array with the `secsgem.secs.variables.SecsVarArray.append()` method.

3.4.1.6 SecsVarList

`secsgem.secs.variables.SecsVarList` is a special type for a list of the different types. The items of the list can be accessed like properties of the object.

An ordered dictionary is required for the creation, because python's default dictionary will be randomly sorted. Sorting is essential because both peers need to have the data in the same order.

```
>>> v=secsgem.SecsVarList([secsgem.OBJACK, secsgem.SOFTREV])
>>> v.OBJACK=3
>>> v.SOFTREV="Hallo"
>>> v
<L [2]
<U1 3 >
<A "Hallo">
```

```
> >>> v.SOFTREV <A "Hallo"> >>> secsgem.format_hex(v.encode())
'01:02:a5:01:03:41:05:48:61:6c:6c:6f'
```

3.4.1.7 SecsVarDynamic

`secsgem.secs.variables.SecsVarDynamic` can take different types, if specified to a certain set of types.

```
>>> v=secsgem.SecsVarDynamic([secsgem.SecsVarString, secsgem.SecsVarU1])
>>> v.set(secsgem.SecsVarString("Hello"))
>>> v
<A "Hello">
>>> v.set(secsgem.SecsVarU1(10))
>>> v
<U1 10 >
>>> v.set(secsgem.SecsVarU4(10))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/home/ext005207/Development/secsgem/secsgem/secs/variables.py", line 255, in _
    ↪set
      raise ValueError("Unsupported type {} for this instance of SecsVarDynamic, _
    ↪allowed {}".format(value.__class__.__name__, self.types))
ValueError: Unsupported type SecsVarU4 for this instance of SecsVarDynamic, allowed [
    ↪<class 'secsgem.secs.variables.SecsVarString'>, <class 'secsgem.secs.variables.
    ↪SecsVarU1'>]
```

3.4.2 Functions

A function is inherited from `secsgem.secs.functionbase.SecsStreamFunction`.

Example:

```
class SecsS02F33(SecsStreamFunction):
    _stream = 2
    _function = 33

    _dataFormat = [
        DATAID,
        [
            [
                RPTID,
                [VID]
            ]
        ]
    ]

    _toHost = False
    _toEquipment = True

    _hasReply = True
    _isReplyRequired = True

    _isMultiBlock = True
```

The data of a function can be read and manipulated with the same functionality as the variables. `secsgem.secs.functionbase.SecsStreamFunction.set()`, `secsgem.secs.functionbase.SecsStreamFunction.get()`, `secsgem.secs.functionbase.SecsStreamFunction.append()`, the index operator and object properties. The objects can also en- and decode themselves.

Usage:

```

>>> f=secsgem.SecsS02F33()
>>> f.DATAID=10
>>> f.DATA.append({"RPTID": 5, "VID": ["Hello", "Hallo"]})
>>> f.DATA.append({"RPTID": 6, "VID": ["1", "2"]})
>>> f
S2F33 W
<L [2]
  <U1 10 >
  <L [2]
    <L [2]
      <U1 5 >
      <L [2]
        <A "Hello">
        <A "Hallo">
      >
    >
  <L [2]
    <U1 6 >
    <L [2]
      <A "1">
      <A "2">
    >
  >
> .
>>> f.DATA[1].VID[0]="Goodbye"
>>> f.DATA[1].VID[1]="Auf Wiedersehen"
>>> f
S2F33 W
<L [2]
  <U1 10 >
  <L [2]
    <L [2]
      <U1 5 >
      <L [2]
        <A "Hello">
        <A "Hallo">
      >
    >
  <L [2]
    <U1 6 >
    <L [2]
      <A "Goodbye">
      <A "Auf Wiedersehen">
    >
  >
> .
>>> secsgem.format_hex(f.encode())

↪ '01:02:a5:01:0a:01:02:01:02:a5:01:05:01:02:41:05:48:65:6c:6c:6f:41:05:48:61:6c:6c:6f:01:02:a5:01:0
↪ '

```

The encoded data can be used as data string in a `secsgem.hsms.packets.HsmsPacket` together with a `secsgem.hsms.packets.HsmsStreamFunctionHeader`. See *Packets*.

3.4.3 Handler

`secsgem.secs.handler.SecsHandler` inherits the functionality from `secsgem.hsms.handler.HsmsHandler` (see *Handler*).

The SecsHandler has additional functionality to add callbacks for specific streams and functions.

```
>>> def s01f13_handler(connection, packet):
...     print "S1F13 received"
...
>>> def on_connect(event, data):
...     print "Connected"
...
>>> client = secsgem.SecsHandler("10.211.55.33", 5000, False, 0, "test")
>>> client.events.hsms_connected += on_connect
>>> client.register_stream_function(1, 13, s01f13_handler)
>>>
>>> client.enable()
Connected
S1F13 received
>>> client.disable()
```

There is also additional functionality concerning collection events, service variables and equipment constants.

3.5 HSMS

SEMI E37

HSMS defines the communication between host and equipment over the TCP protocol. It specifies packets used to initiate and terminate the connection, check if the link is still active and transfer the actual data.

3.5.1 Packets

A HSMS packet `secsgem.hsms.packets.HsmsPacket` consists of a header `secsgem.hsms.packets.HsmsHeader` and a data part represented by a string. The string contains the additional data encoded as ASCII characters for transmission over TCP. The additional data is only required for a stream/function packet.

```
>>> secsgem.hsms.packets.HsmsPacket(secsgem.hsms.packets.HsmsLinktestReqHeader(2))
secsgem.hsms.packets.HsmsPacket({'header': secsgem.hsms.packets.HsmsLinktestReqHeader(
↳ {'function': 0, 'stream': 0, 'pType': 0, 'system': 2, 'sessionID': 65535,
↳ 'requireResponse': False, 'sType': 5}), 'data': ''})
```

Every header has a system id to match the response to a certain request. The system id is the first parameter to the headers constructor. The connection keeps track of the system id, a new one can be requested with the `secsgem.hsms.connections.HsmsConnection.get_next_system_counter()` function.

HSMS packet objects can encode themselves with the `secsgem.hsms.packets.HsmsPacket.encode()` function to a string, which can be sent over the TCP connection.

```
>>> packet = secsgem.hsms.packets.HsmsPacket(secsgem.hsms.packets.
↳ HsmsLinktestReqHeader(2))
>>> secsgem.common.format_hex(packet.encode())
'00:00:00:0a:ff:ff:00:00:00:05:00:00:00:02'
```

The other way around, a HSMS packet object can be created from the ASCII string with the `secsgem.hsms.packets.HsmsPacket.decode()` function.

```
>>> secsgem.hsms.packets.HsmsPacket.decode(packetData)
secsgem.hsms.packets.HsmsPacket({'header': secsgem.hsms.packets.HsmsHeader({'function': 0, 'stream': 0, 'pType': 0, 'system': 2, 'sessionID': 65535, 'requireResponse': False, 'sType': 5}), 'data': ''})
```

There are classes inherited from `secsgem.hsms.packets.HsmsHeader` for all HSMS packet types available:

Type	Class	SType
Select Request	<code>secsgem.hsms.packets.HsmsSelectReqHeader</code>	1
Select Response	<code>secsgem.hsms.packets.HsmsSelectRspHeader</code>	2
Deselect Request	<code>secsgem.hsms.packets.HsmsDeselectReqHeader</code>	3
Deselect Response	<code>secsgem.hsms.packets.HsmsDeselectRspHeader</code>	4
Linktest Request	<code>secsgem.hsms.packets.HsmsLinktestReqHeader</code>	5
Linktest Response	<code>secsgem.hsms.packets.HsmsLinktestRspHeader</code>	6
Reject Request	<code>secsgem.hsms.packets.HsmsRejectReqHeader</code>	7
Separate Request	<code>secsgem.hsms.packets.HsmsSeparateReqHeader</code>	9
Data Message	<code>secsgem.hsms.packets.HsmsStreamFunctionHeader</code>	0

3.5.1.1 Select Request

Establish HSMS communication

```
>>> secsgem.hsms.packets.HsmsSelectReqHeader(14)
secsgem.hsms.packets.HsmsSelectReqHeader({'function': 0, 'stream': 0, 'pType': 0, 'system': 14, 'sessionID': 65535, 'requireResponse': False, 'sType': 1})
```

3.5.1.2 Select Response

Result of select request

```
>>> secsgem.hsms.packets.HsmsSelectRspHeader(24)
secsgem.hsms.packets.HsmsSelectRspHeader({'function': 0, 'stream': 0, 'pType': 0, 'system': 24, 'sessionID': 65535, 'requireResponse': False, 'sType': 2})
```

3.5.1.3 Deselect Request

Grateful close HSMS communication before disconnecting

```
>>> secsgem.hsms.packets.HsmsDeselectReqHeader(1)
secsgem.hsms.packets.HsmsDeselectReqHeader({'function': 0, 'stream': 0, 'pType': 0, 'system': 1, 'sessionID': 65535, 'requireResponse': False, 'sType': 3})
```

3.5.1.4 Deselect Response

Result of deselect request


```
>>> secsgem.hsms.packets.HsmsDeselectRspHeader(1)
secsgem.hsms.packets.HsmsDeselectRspHeader({'function': 0, 'stream': 0, 'pType': 0,
↪ 'system': 1, 'sessionID': 65535, 'requireResponse': False, 'sType': 4})
```

3.5.1.5 Linktest Request

Check the HSMS connection link is good

```
>>> secsgem.hsms.packets.HsmsLinktestReqHeader(2)
secsgem.hsms.packets.HsmsLinktestReqHeader({'function': 0, 'stream': 0, 'pType': 0,
↪ 'system': 2, 'sessionID': 65535, 'requireResponse': False, 'sType': 5})
```

3.5.1.6 Linktest Response

Result of linktest request

```
>>> secsgem.hsms.packets.HsmsLinktestRspHeader(10)
secsgem.hsms.packets.HsmsLinktestRspHeader({'function': 0, 'stream': 0, 'pType': 0,
↪ 'system': 10, 'sessionID': 65535, 'requireResponse': False, 'sType': 6})
```

3.5.1.7 Reject Request

Response to unsupported HSMS message

```
>>> secsgem.hsms.packets.HsmsRejectReqHeader(17, 3, 4)
secsgem.hsms.packets.HsmsRejectReqHeader({'function': 4, 'stream': 3, 'pType': 0,
↪ 'system': 17, 'sessionID': 65535, 'requireResponse': False, 'sType': 7})
```

3.5.1.8 Separate Request

Immediate termination of the HSMS connection

```
>>> secsgem.hsms.packets.HsmsSeparateReqHeader(17)
secsgem.hsms.packets.HsmsSeparateReqHeader({'function': 0, 'stream': 0, 'pType': 0,
↪ 'system': 17, 'sessionID': 65535, 'requireResponse': False, 'sType': 9})
```

3.5.1.9 Data Message

Secs stream and function message

```
>>> secsgem.hsms.packets.HsmsStreamFunctionHeader(22, 1, 1, True, 100)
secsgem.hsms.packets.HsmsStreamFunctionHeader({'function': 1, 'stream': 1, 'pType': 0,
↪ 'system': 22, 'sessionID': 100, 'requireResponse': True, 'sType': 0})
```

3.5.2 Connections

HSMS has active and passive connections. The active connection is the one making the connection, the passive one is waiting for the incoming connection.

The implementation for the active connection is `secsgem.hsms.connections.HsmsActiveConnection`. For the passive connection there are two implementations:

- `secsgem.hsms.connections.HsmsPassiveConnection` handles only one connection at a time.
- `secsgem.hsms.connections.HsmsMultiPassiveConnection` together with `secsgem.hsms.connections.HsmsMultiPassiveServer` handle multiple connections from different peers.

All connection classes are based on the `secsgem.hsms.connections.HsmsConnection` class, which provides common functionality for all connection types.

The connection process for active and passive connections can be started with the `secsgem.hsms.connections.HsmsPassiveConnection.enable()` function, and stopped with the `secsgem.hsms.connections.HsmsPassiveConnection.disable()` function.

3.5.2.1 Delegates

All connections work with delegates. When a connection is established/terminated or a packet is received a method of the passed delegate object will be called. The connections support the following delegates:

- `on_connection_established(connection)`
- `on_connection_packet_received(response)`
- `on_connection_before_closed(connection)`
- `on_connection_closed(connection)`

Sample delegate class:

```
class DelegateSample:
    def on_connection_established(self, connection):
        print "Connection established"

    def on_connection_packet_received(self, connection, packet):
        print "Packet received", packet

    def on_connection_before_closed(self, connection):
        print "Connection about to be terminated"

    def on_connection_closed(self, connection):
        print "Connection terminated"
```

3.5.2.2 Active connection

For the active connection the first parameter is the IP address of the peer, the second parameter is the port of the peer. The third parameter is the session id the peer is configured for.

Example:

```
>>> delegate = DelegateSample()
>>> conn = secsgem.HsmsActiveConnection('10.211.55.33', 5000, 0, delegate)
>>> conn.enable()
Connection established
Packet received header: {sessionID:0x0000, stream:00, function:04, pType:0x00,
↪sType:0x07, system:0x00000000, requireResponse:0}
Packet received header: {sessionID:0x0000, stream:00, function:01, pType:0x00,
↪sType:0x07, system:0x00000000, requireResponse:0}
```

(continues on next page)

(continued from previous page)

```

Connection about to be terminated
Connection terminated
>>> conn.disable()

```

3.5.2.3 Passive connection

For the passive connection the first parameter is the expected IP address of the peer, the second parameter is the port to listen on. The third parameter is the session id the peer is configured for.

Example:

```

>>> delegate = DelegateSample()
>>> conn = secsgem.HsmsPassiveConnection('10.211.55.33', 5000, 0, delegate)
>>> conn.enable()
Connection established
Packet received header: {sessionID:0xffff, stream:00, function:00, pType:0x00,
↪sType:0x01, system:0x00000001, requireResponse:0}
Packet received header: {sessionID:0x0000, stream:00, function:03, pType:0x00,
↪sType:0x07, system:0x00000000, requireResponse:0}
Connection about to be terminated
Connection terminated
>>> conn.disable()

```

3.5.2.4 Multi-passive connection

In this mode one listening port handles the incoming connections for more than one peer. A instance of *secsgem.hsms.connections.HsmsMultiPassiveServer* is created and connection is created using its *secsgem.hsms.connections.HsmsMultiPassiveServer.create_connection()* method. The parameters of the method are the same as for the *Passive connection*. For every available peer a connection must be created using this method.

Example:

```

>>> delegate = DelegateSample()
>>> server = secsgem.HsmsMultiPassiveServer(5000)
>>> conn = server.create_connection('10.211.55.33', 5000, 0, delegate)
>>> conn.enable()
>>> server.start()
Connection established
Packet received header: {sessionID:0xffff, stream:00, function:00, pType:0x00,
↪sType:0x01, system:0x00000003, requireResponse:0}
Packet received header: {sessionID:0x0000, stream:00, function:03, pType:0x00,
↪sType:0x07, system:0x00000000, requireResponse:0}
Connection about to be terminated
Connection terminated
>>> conn.disable()
>>> server.stop()

```

3.5.2.5 Connection manager

The *secsgem.hsms.connectionmanager.HsmsConnectionManager* can be used to manage multiple active and passive connections. It creates and removes *secsgem.hsms.connections.HsmsActiveConnection* and *secsgem.hsms.connections.*

HsmsMultiPassiveServer/secsgem.hsms.connections.HsmsMultiPassiveConnection dynamically.

```
>>> manager=secsgem.HsmsConnectionManager()
>>> handler=manager.add_peer("connection", '10.211.55.33', 5000, False, 0)
>>> handler.enable()
>>> handler.send_linktest_req()
secsgem.hsms.packets.HsmsPacket({'header': secsgem.hsms.packets.HsmsHeader({'function
↪': 0, 'stream': 0, 'pType': 0, 'system': 13, 'sessionID': 65535, 'requireResponse':
↪False, 'sType': 6}), 'data': ''})
>>> handler.disable()
>>> manager.stop()
```

Connection manager works with *handlers* which take care of a lot of the required communication on the matching level (*secsgem.hsms.handler.HsmsHandler*, *secsgem.secs.handler.SecsHandler* and *secsgem.gem.handler.GemHandler*).

3.5.3 Handler

secsgem.hsms.handler.HsmsHandler has the basic HSMS connection handling build in. It automatically selects and deselects the link and performs a periodic linktest. It also replies to incoming HSMS requests like linktest automatically.

```
>>> def on_connect(event, data):
...     print "Connected"
...
>>> client = secsgem.HsmsHandler("10.211.55.33", 5000, False, 0, "test")
>>> client.events.hsms_connected += on_connect
>>> client.enable()
Connected
>>> client.send_linktest_req()
secsgem.hsms.packets.HsmsPacket({'header': secsgem.hsms.packets.HsmsHeader({'function
↪': 0, 'stream': 0, 'pType': 0, 'system': 7, 'sessionID': 65535, 'requireResponse':
↪False, 'sType': 6}), 'data': ''})
>>> client.disable()
```

The handler has functions to send requests and responses and wait for a certain response.

3.5.3.1 Events

Events of the handler can be received with the help of *secsgem.common.EventHandler*. The handler sends the following events:

Event name	Description
hsms_connected	Connection was established
hsms_selected	Connection was selected
hsms_disconnected	Connection was terminated

For an example on how to use these events see the code fragment above.

3.6 Class reference

3.6.1 HSMS

3.6.1.1 Packets

class `secsgem.hsms.packets.HsmsPacket` (*header=None, data=b''*)

Bases: `object`

Class for hsms packet.

Contains all required data and functions.

encode ()

Encode packet data to hsms packet.

Returns encoded packet

Return type `string`

Example:

```
>>> import secsgem
>>>
>>> packet = secsgem.hsms.packets.HsmsPacket(secsgem.hsms.packets.
↳ HsmsLinktestReqHeader(2))
>>> secsgem.common.format_hex(packet.encode())
'00:00:00:0a:ff:ff:00:00:00:05:00:00:00:02'
```

static decode (*text*)

Decode byte array hsms packet to HsmsPacket object.

Returns received packet object

Return type `secsgem.hsms.packets.HsmsPacket`

Example:

```
>>> import secsgem
>>>
>>> packetData = b"\x00\x00\x00\x0b\xff\xff\x00\x00\x00\x05\x00\x00\x00\x02"
>>>
>>> secsgem.format_hex(packetData)
'00:00:00:0b:ff:ff:00:00:00:05:00:00:00:02'
>>>
>>> secsgem.hsms.packets.HsmsPacket.decode(packetData)
HsmsPacket({'header': HsmsHeader({'sessionID:0xffff, stream:00, function:00,
↳ pType:0x00, sType:0x05, system:0x00000002, requireResponse:False}), 'data':
↳ ''})
```

class `secsgem.hsms.packets.HsmsHeader` (*system, session_id*)

Bases: `object`

Generic HSMS header.

Base for different specific headers

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsStreamFunctionHeader**(system, stream, function, require_response, session_id)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for SECS message.

Header for message with SType 0.

encode()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsSelectReqHeader**(system)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Select Request.

Header for message with SType 1.

encode()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsSelectRspHeader**(system)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Select Response.

Header for message with SType 2.

encode()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsDeselectReqHeader** (*system*)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Deselect Request.

Header for message with SType 3.

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsDeselectRspHeader** (*system*)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Deselect Response.

Header for message with SType 4.

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsLinktestReqHeader** (*system*)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Linktest Request.

Header for message with SType 5.

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsLinktestRspHeader** (system)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Linktest Response.

Header for message with SType 6.

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsRejectReqHeader** (system, s_type, reason)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Reject Request.

Header for message with SType 7.

encode ()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

class secsgem.hsms.packets.**HsmsSeparateReqHeader** (system)

Bases: *secsgem.hsms.packets.HsmsHeader*

Header for Separate Request.

Header for message with SType 9.

encode()

Encode header to hsms packet.

Returns encoded header

Return type string

Example:

```
>>> import secsgem
>>>
>>> header = secsgem.hsms.packets.HsmsLinktestReqHeader(2)
>>> secsgem.common.format_hex(header.encode())
'ff:ff:00:00:00:05:00:00:00:02'
```

3.6.1.2 Connections

class secsgem.hsms.connections.**HsmsConnection** (*active, address, port, session_id=0, delegate=None*)

Bases: object

Connection class used for active and passive hsms connections.

selectTimeout = 0.5

Timeout for select calls .

sendBlockSize = 1048576

Block size for outbound data .

T3 = 45.0

Reply Timeout .

T5 = 10.0

Connect Separation Time .

T6 = 5.0

Control Transaction Timeout .

disconnect()

Close connection.

send_packet (*packet*)

Send the ASCII coded packet to the remote host.

Parameters **packet** (*string / byte array*) – encoded data to be transmitted

class secsgem.hsms.connections.**HsmsActiveConnection** (*address, port=5000, session_id=0, delegate=None*)

Bases: *secsgem.hsms.connections.HsmsConnection*

Client class for single active (outgoing) connection.

T3 = 45.0

T5 = 10.0

T6 = 5.0

disconnect()

Close connection.

selectTimeout = 0.5

sendBlockSize = 1048576

send_packet (*packet*)

Send the ASCII coded packet to the remote host.

Parameters **packet** (*string* / *byte array*) – encoded data to be transmitted

enable ()

Enable the connection.

Starts the connection process to the passive remote.

disable ()

Disable the connection.

Stops all connection attempts, and closes the connection

class secsgem.hsms.connections.**HsmsPassiveConnection** (*address*, *port*=5000, *session_id*=0, *delegate*=None)

Bases: *secsgem.hsms.connections.HsmsConnection*

Server class for single passive (incoming) connection.

Creates a listening socket and waits for one incoming connection on this socket. After the connection is established the listening socket is closed.

enable ()

Enable the connection.

Starts the connection process to the passive remote.

disable ()

Disable the connection.

Stops all connection attempts, and closes the connection

T3 = 45.0

T5 = 10.0

T6 = 5.0

disconnect ()

Close connection.

selectTimeout = 0.5

sendBlockSize = 1048576

send_packet (*packet*)

Send the ASCII coded packet to the remote host.

Parameters **packet** (*string* / *byte array*) – encoded data to be transmitted

class secsgem.hsms.connections.**HsmsMultiPassiveConnection** (*address*, *port*=5000, *session_id*=0, *delegate*=None)

Bases: *secsgem.hsms.connections.HsmsConnection*

Connection class for single connection from *secsgem.hsms.connections.HsmsMultiPassiveServer*.

Handles connections incoming connection from *secsgem.hsms.connections.HsmsMultiPassiveServer*

on_connected (*sock*, *address*)

Connected callback for *secsgem.hsms.connections.HsmsMultiPassiveServer*.

Parameters

- **sock** (*Socket*) – Socket for new connection
- **address** (*string*) – IP address of remote host

enable ()

Enable the connection.

Starts the connection process to the passive remote.

disable ()

Disable the connection.

Stops all connection attempts, and closes the connection

T3 = 45.0

T5 = 10.0

T6 = 5.0

disconnect ()

Close connection.

selectTimeout = 0.5

sendBlockSize = 1048576

send_packet (*packet*)

Send the ASCII coded packet to the remote host.

Parameters packet (*string / byte array*) – encoded data to be transmitted

class secsgem.hsms.connections.HsmsMultiPassiveServer (*port=5000*)

Bases: object

Server class for multiple passive (incoming) connection.

The server creates a listening socket and waits for incoming connections on this socket.

selectTimeout = 0.5

Timeout for select calls .

create_connection (*address, port=5000, session_id=0, delegate=None*)

Create and remember connection for the server.

Parameters

- **address** (*string*) – IP address of target host
- **port** (*integer*) – TCP port of target host
- **session_id** (*integer*) – session / device ID to use for connection
- **delegate** (*object*) – target for messages

start ()

Starts the server and returns.

It will launch a listener running in background to wait for incoming connections.

stop (*terminate_connections=True*)

Stops the server. The background job waiting for incoming connections will be terminated.

Optionally all connections received will be closed.

Parameters **terminate_connections** (*boolean*) – terminate all connection made by this server

3.6.1.3 Handler

class secsgem.hsms.handler.HsmsHandler (*address, port, active, session_id, name, custom_connection_handler=None*)

Bases: object

Baseclass for creating Host/Equipment models.

This layer contains the HSMS functionality. Inherit from this class and override required functions.

events

Property for event handling.

callbacks

Property for callback handling.

get_next_system_counter ()

Returns the next System.

Returns System for the next command

Return type integer

on_connection_established (_)

Connection was established.

on_connection_before_closed (_)

Connection is about to be closed.

on_connection_closed (_)

Connection was closed.

on_connection_packet_received (_, packet)

Packet received by connection.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – received data packet

enable ()

Enables the connection.

disable ()

Disables the connection.

send_stream_function (packet)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

send_and_waitfor_response (packet)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

Returns Packet that was received

Return type *secsgem.hsms.packets.HsmsPacket*

send_response (function, system)

Send response function for system.

Parameters

- **function** (*secsgem.secs.functionbase.SecsStreamFunction*) – function to be sent
- **system** (*integer*) – system to reply to

send_select_req()

Send a Select Request to the remote host.

Returns System of the sent request**Return type** integer**send_select_rsp** (*system_id*)

Send a Select Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for**send_linktest_req()**

Send a Linktest Request to the remote host.

Returns System of the sent request**Return type** integer**send_linktest_rsp** (*system_id*)

Send a Linktest Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for**send_deselect_req()**

Send a Deselect Request to the remote host.

Returns System of the sent request**Return type** integer**send_deselect_rsp** (*system_id*)

Send a Deselect Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for**send_reject_rsp** (*system_id, s_type, reason*)

Send a Reject Response to the remote host.

Parameters

- **system_id** (*integer*) – System of the request to reply for
- **s_type** (*integer*) – s_type of rejected message
- **reason** (*integer*) – reason for rejection

send_separate_req()

Send a Separate Request to the remote host.

3.6.1.4 ConnectionManager**class** secsgem.hsms.connectionmanager.HsmsConnectionManager

Bases: object

High level class that handles multiple active and passive connections and the model for them.

events

Property for event handling.

has_connection_to (*index*)

Check if connection to certain peer exists.

Parameters **index** (*string*) – Name of the requested handler.

Returns Is peer available

Return type boolean

static get_connection_id (*address*)

Generates connection ids used for internal indexing.

Parameters **address** (*string*) – The IP address for the affected remote.

add_peer (*name, address, port, active, session_id, connection_handler=<class 'secsgem.hsms.handler.HsmsHandler'>*)

Add a new connection.

Parameters

- **name** (*string*) – Name of the peers configuration
- **address** (*string*) – IP address of peer
- **port** (*integer*) – TCP port of peer
- **active** (*boolean*) – Is the connection active (*True*) or passive (*False*)
- **session_id** (*integer*) – session / device ID of peer
- **connection_handler** (inherited from `secsgem.hsms.handler.HsmsHandler`) – Model handling this connection

remove_peer (*name, address, port*)

Remove a previously added connection.

Parameters

- **name** (*string*) – Name of the peers configuration
- **address** (*string*) – IP address of peer
- **port** (*integer*) – TCP port of peer

stop ()

Stop all servers and terminate the connections.

3.6.2 SECS

3.6.2.1 Variables

SECS variable types.

class `secsgem.secs.variables.SecsVar`

Bases: `object`

Base class for SECS variables.

Due to the python types, wrapper classes for variables are required. If constructor is called with `SecsVar` or subclass only the value is copied.

formatCode = -1

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static **get_format** (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*various*) – new value

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

class `secsgem.secs.variables.SecsVarDynamic` (*types, value=None, count=-1*)

Bases: `secsgem.secs.variables.SecsVar`

Variable with interchangeable type.

set (*value*)

Set the internal value to the provided value.

In doubt provide the variable wrapped in the matching `secsgem.secs.variables.SecsVar` class, to avoid confusion.

Example:

```
>>> import secsgem
>>>
>>> var = secsgem.SecsVarDynamic([secsgem.SecsVarString, secsgem.SecsVarU1])
>>> var.set(secsgem.SecsVarU1(10))
>>> var
<U1 10 >
```

If no type is provided the default type is used which might not be the expected type.

Parameters **value** (*various*) – new value

get ()

Return the internal value.

Returns internal value

Return type various

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (data, start=0)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (data, text_pos=0)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (length)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

formatCode = -1

static generate (dataformat)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static get_format (dataformat, showname=False)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

class secsgem.secs.variables.**ANYVALUE** (value=None)

Bases: *secsgem.secs.variables.SecsVarDynamic*

Dummy data item for generation of unknown types.

Types

- *SecsVarArray*
- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

formatCode = -1

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type various

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

In doubt provide the variable wrapped in the matching `secsgem.secs.variables.SecsVar` class, to avoid confusion.

Example:

```
>>> import secsgem
>>>
>>> var = secsgem.SecsVarDynamic([secsgem.SecsVarString, secsgem.SecsVarU1])
>>> var.set(secsgem.SecsVarU1(10))
>>> var
<U1 10 >
```

If no type is provided the default type is used which might not be the expected type.

Parameters **value** (*various*) – new value

class `secsgem.secs.variables.SecsVarList` (*dataformat, value=None*)

Bases: `secsgem.secs.variables.SecsVar`

List variable type. List with items of different types.

formatCode = 0

textCode = 'L'

preferredTypes = [`<class 'dict'>`]

static get_format (*dataformat, showname=False*)

Gets the format of the variable.

Returns returns the string representation of the function

Return type string

static get_name_from_format (*dataformat*)

Generates a name for the passed dataformat.

Parameters `dataformat` (*list/SecsVar based class*) – dataformat to get name for

Returns name for dataformat

Return type str

set (*value*)

Set the internal value to the provided value.

Parameters `value` (*dict/list*) – new value

get ()

Return the internal value.

Returns internal value

Return type list

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters `length` (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters `dataformat` (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

class secsgem.secs.variables.SecsVarArray (*dataFormat*, *value=None*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVar*

List variable type. List with items of same type.

formatCode = 0

textCode = 'L'

preferredTypes = [*<class 'list'>*]

static get_format (*dataformat*, *showname=False*)

Gets the format of the variable.

Returns returns the string representation of the function

Return type string

append (*data*)

Append data to the internal list.

Parameters **value** (*various*) – new value

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list*) – new value

get ()

Return the internal value.

Returns internal value

Return type list

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

class secsgem.secs.variables.**SecsVarBinary** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVar*

Secs type for binary data.

formatCode = 8

textCode = 'B'

preferredTypes = [*<class 'bytes'>*, *<class 'bytearray'>*]

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*string/integer*) – new value

get ()

Return the internal value.

Returns internal value

Return type list/integer

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos*=0)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static get_format (*dataformat*, *showname*=False)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

class secsgem.secs.variables.**SecsVarBoolean** (*value*=None, *count*=-1)

Bases: *secsgem.secs.variables.SecsVar*

Secs type for boolean data.

formatCode = 9

textCode = 'BOOLEAN'

preferredTypes = [<class 'bool'>]

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/boolean*) – new value

get ()

Return the internal value.

Returns internal value

Return type list/boolean

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (data, start=0)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (data, text_pos=0)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (length)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (dataformat)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static get_format (dataformat, showname=False)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

class secsgem.secs.variables.**SecsVarText** (*value=""*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVar*

Secs type base for any text data.

formatCode = -1

textCode = ''

controlChars = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12

coding = ''

supports_value (*value*)
 Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

set (*value*)
 Set the internal value to the provided value.

Parameters **value** (*string/integer*) – new value

get ()
 Return the internal value.

Returns internal value

Return type string

encode ()
 Encode the value to secs data.

Returns encoded data bytes

Return type string

decode (*data, start=0*)
 Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)
 Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode_item_header (*length*)
 Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)
 Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

class secsgem.secs.variables.**SecsVarString** (*value=""*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVarText*

Secs type for string data.

Parameters

- **value** (*string*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 16

textCode = 'A'

preferredTypes = [*<class 'bytes'>*, *<class 'str'>*]

controlChars = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12'

coding = 'latin-1'

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type string

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*string/integer*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarJIS8** (*value=”, count=-1*)

Bases: *secsgem.secs.variables.SecsVarText*

Secs type for string data.

Parameters

- **value** (*string*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 17

textCode = 'J'

preferredTypes = [**<class 'bytes'>**, **<class 'str'>**]

controlChars = '\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12

coding = 'jis-8'

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type string

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*string/integer*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarNumber** (*value=None*, *count=-1*)

Bases: [secsgem.secs.variables.SecsVar](#)

Secs base type for numeric data.

formatCode = 0

textCode = ''

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

get ()

Return the internal value.

Returns internal value

Return type *list/integer/float*

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type *string*

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type *integer*

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (*integer, integer, integer*)

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type *string*

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

class secsgem.secs.variables.**SecsVarI8** (*value=None*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 8 byte signed data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 24

textCode = 'I8'

preferredTypes = [*<class 'int'>*]

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarI1** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 1 byte signed data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 25

textCode = 'I1'

preferredTypes = [*<class 'int'>*]

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarI2** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 2 byte signed data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 26

textCode = 'I2'

preferredTypes = [<class 'int'>]

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters *value* (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters *value* (*any*) – value to test

class secsgem.secs.variables.**SecsVarI4** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 4 byte signed data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 28

textCode = 'I4'

preferredTypes = [*<class 'int'>*]

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarF8** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 8 byte float data.

Parameters

- **value** (*list/float*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 32

textCode = 'F8'

preferredTypes = [*<class 'float'>*]

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarF4** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 4 byte float data.

Parameters

- **value** (*list/float*) – initial value

- **count** (*integer*) – number of items this value

formatCode = 36

textCode = 'F4'

preferredTypes = [<class 'float'>]

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarU8** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 8 byte unsigned data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 40

textCode = 'U8'

preferredTypes = [*<class 'int'>*]

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data, text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarU1** (*value=None, count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 1 byte unsigned data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 41

textCode = 'U1'

preferredTypes = [*<class 'int'>*]

decode (*data, start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarU2** (*value=None*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 2 byte unsigned data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 42

textCode = 'U2'

preferredTypes = [<class 'int'>]

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)

Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()

Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat*, *showname=False*)

Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)

Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)

Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

class secsgem.secs.variables.**SecsVarU4** (*value=None*, *count=-1*)

Bases: *secsgem.secs.variables.SecsVarNumber*

Secs type for 4 byte unsigned data.

Parameters

- **value** (*list/integer*) – initial value
- **count** (*integer*) – number of items this value

formatCode = 44

textCode = 'U4'

decode (*data*, *start=0*)

Decode the secs byte data to the value.

Parameters

- **data** (*string*) – encoded data bytes
- **start** (*integer*) – start position of value the data

Returns new start position

Return type integer

decode_item_header (*data*, *text_pos=0*)

Encode item header depending on the number of length bytes required.

Parameters

- **data** (*string*) – encoded data
- **text_pos** (*integer*) – start of item header in data

Returns start position for next item, format code, length item of data

Return type (integer, integer, integer)

encode ()

Encode the value to secs data.

Returns encoded data bytes

Return type string

encode_item_header (*length*)

Encode item header depending on the number of length bytes required.

Parameters **length** (*integer*) – number of bytes in data

Returns encoded item header bytes

Return type string

static generate (*dataformat*)
Generate actual variable from data format.

Parameters **dataformat** (*list/SecsVar based class*) – dataformat to create variable for

Returns created variable

Return type SecsVar based class

get ()
Return the internal value.

Returns internal value

Return type list/integer/float

static get_format (*dataformat, showname=False*)
Gets the format of the function.

Returns returns the string representation of the function

Return type string

set (*value*)
Set the internal value to the provided value.

Parameters **value** (*list/integer/float*) – new value

supports_value (*value*)
Check if the current instance supports the provided value.

Parameters **value** (*any*) – value to test

preferredTypes = [`<class 'int'>`]

3.6.2.2 Data Items

Data items for functions.

class secsgem.secs.dataitems.**DataItemMeta**
Bases: type
Meta class for data items.

class secsgem.secs.dataitems.**DataItemBase** (*value=None*)
Bases: object
Base class for data items.
It provides type and output handling.

classmethod **get_format** (*showname=True*)
Format the contents as a string.

Parameters **showname** – Display the real class name when True

Returns Formatted value string

```
class secsgem.secs.dataitems.ACKC5 (value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Accepted	<i>secsgem.secs.dataitems.ACKC5.ACCEPTED</i>
1-63	Error	<i>secsgem.secs.dataitems.ACKC5.ERROR</i>

Used In Function

- *SecsS05F02*
- *SecsS05F04*

ACCEPTED = 0

ERROR = 1

```
class secsgem.secs.dataitems.ACKC6 (value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Accepted	<i>secsgem.secs.dataitems.ACKC6.ACCEPTED</i>
1-63	Error	<i>secsgem.secs.dataitems.ACKC6.ERROR</i>

Used In Function

- *SecsS06F02*
- *SecsS06F04*
- *SecsS06F10*
- *SecsS06F12*
- *SecsS06F14*

ACCEPTED = 0

ERROR = 1

```
class secsgem.secs.dataitems.ACKC7 (value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Accepted	<i>secsgem.secs.dataitems.ACKC7.ACCEPTED</i>
1	Permission not granted	<i>secsgem.secs.dataitems.ACKC7.NO_PERMISSION</i>
2	Length error	<i>secsgem.secs.dataitems.ACKC7.LENGTH_ERROR</i>
3	Matrix overflow	<i>secsgem.secs.dataitems.ACKC7.MATRIX_OVERFLOW</i>
4	PPID not found	<i>secsgem.secs.dataitems.ACKC7.PPID_NOT_FOUND</i>
5	Mode unsupported	<i>secsgem.secs.dataitems.ACKC7.MODE_UNSUPPORTED</i>
6	Performed later	<i>secsgem.secs.dataitems.ACKC7.PERFORMED_LATER</i>
7-63	Reserved	

Used In Function

- *SecsS07F04*
- *SecsS07F12*
- *SecsS07F14*
- *SecsS07F16*
- *SecsS07F18*
- *SecsS07F24*
- *SecsS07F32*
- *SecsS07F38*
- *SecsS07F40*
- *SecsS07F42*
- *SecsS07F44*

ACCEPTED = 0

NO_PERMISSION = 1

LENGTH_ERROR = 2

MATRIX_OVERFLOW = 3

PPID_NOT_FOUND = 4

MODE_UNSUPPORTED = 5

PERFORMED_LATER = 6

class *secsgem.secs.dataitems.ACKC10* (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Acknowledge code.

Types *SecsVarBinary*

Length 1**Values**

Value	Description	Constant
0	Accepted	<code>secsgem.secs.dataitems.ACKC10.ACCEPTED</code>
1	Will not be displayed	<code>secsgem.secs.dataitems.ACKC10.NOT_DISPLAYED</code>
2	Terminal not available	<code>secsgem.secs.dataitems.ACKC10.TERMINAL_NOT_AVAILABLE</code>
3-63	Other error	

Used In Function

- `SecsS10F02`
- `SecsS10F04`
- `SecsS10F06`
- `SecsS10F10`

ACCEPTED = 0**NOT_DISPLAYED = 1****TERMINAL_NOT_AVAILABLE = 2****class** `secsgem.secs.dataitems.ACKA` (*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarBoolean`

Request success.

Types `SecsVarBoolean`**Length 1****Values**

Value	
True	Success
False	Failed

Used In Function

- `SecsS05F14`
- `SecsS05F15`
- `SecsS05F18`
- `SecsS16F02`
- `SecsS16F04`
- `SecsS16F06`
- `SecsS16F12`
- `SecsS16F14`

- SecsS16F16
- SecsS16F18
- SecsS16F24
- SecsS16F26
- SecsS16F28
- SecsS16F30
- SecsS17F04
- SecsS17F08
- SecsS17F14

class secsgem.secs.dataitems.**ALCD** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Alarm code byte.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Not used	
1	Personal safety	<i>secsgem.secs.dataitems.ALCD.PERSONAL_SAFETY</i>
2	Equipment safety	<i>secsgem.secs.dataitems.ALCD.EQUIPMENT_SAFETY</i>
3	Parameter control warning	<i>secsgem.secs.dataitems.ALCD.PARAMETER_CONTROL_WARNING</i>
4	Parameter control error	<i>secsgem.secs.dataitems.ALCD.PARAMETER_CONTROL_ERROR</i>
5	Irrecoverable error	<i>secsgem.secs.dataitems.ALCD.IRRECOVERABLE_ERROR</i>
6	Equipment status warning	<i>secsgem.secs.dataitems.ALCD.EQUIPMENT_STATUS_WARNING</i>
7	Attention flags	<i>secsgem.secs.dataitems.ALCD.ATTENTION_FLAGS</i>
8	Data integrity	<i>secsgem.secs.dataitems.ALCD.DATA_INTEGRITY</i>
9-63	Other catogories	
128	Alarm set flag	<i>secsgem.secs.dataitems.ALCD.ALARM_SET</i>

Used In Function

- *SecsS05F01*
- *SecsS05F06*

PERSONAL_SAFETY = 1

EQUIPMENT_SAFETY = 2

PARAMETER_CONTROL_WARNING = 3

PARAMETER_CONTROL_ERROR = 4

IRRECOVERABLE_ERROR = 5

EQUIPMENT_STATUS_WARNING = 6

ATTENTION_FLAGS = 7

DATA_INTEGRITY = 8

ALARM_SET = 128

class secsgem.secs.dataitems.**ALED** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Alarm en-/disable code byte.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Disable	<i>secsgem.secs.dataitems.ALED.DISABLE</i>
1-127	Not used	
128	Enable	<i>secsgem.secs.dataitems.ALED.ENABLE</i>
129-255	Not used	

Used In Function

- *SecsS05F03*

DISABLE = 0

ENABLE = 128

class secsgem.secs.dataitems.**ALID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Alarm ID.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS05F01*
- *SecsS05F03*
- *SecsS05F05*

- *SecsS05F06*

class secsgem.secs.dataitems.**ALTX**(*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Alarm ID.

Types

- *SecsVarString*

Used In Function

- *SecsS05F01*
- *SecsS05F06*

class secsgem.secs.dataitems.**ATTRDATA**(*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Object attribute value.

Types

- *SecsVarArray*
- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS01F20*
- *SecsS03F17*
- *SecsS03F18*
- *SecsS13F14*
- *SecsS13F16*
- *SecsS14F01*

- *SecsS14F02*
- *SecsS14F03*
- *SecsS14F04*
- *SecsS14F09*
- *SecsS14F10*
- *SecsS14F11*
- *SecsS14F12*
- *SecsS14F13*
- *SecsS14F14*
- *SecsS14F15*
- *SecsS14F16*
- *SecsS14F17*
- *SecsS14F18*
- *SecsS18F02*
- *SecsS18F03*

class secsgem.secs.dataitems.**ATTRID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Object attribute identifier.

Types

- *SecsVarString*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS01F19*
- *SecsS03F17*
- *SecsS03F18*
- *SecsS13F14*
- *SecsS13F16*
- *SecsS14F01*
- *SecsS14F02*
- *SecsS14F03*
- *SecsS14F04*
- *SecsS14F08*

- SecsS14F09
- SecsS14F10
- SecsS14F11
- SecsS14F12
- SecsS14F13
- SecsS14F14
- SecsS14F15
- SecsS14F16
- SecsS14F17
- SecsS14F18
- SecsS18F01
- SecsS18F03

class secsgem.secs.dataitems.**ATTRRELN** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU1*

Attribute relation to attribute of object.

Types *SecsVarU1*

Values

Value	Description	Constant
0	Equal to	<i>secsgem.secs.dataitems.ATTRRELN.EQUAL</i>
1	Not equal to	<i>secsgem.secs.dataitems.ATTRRELN.NOT_EQUAL</i>
2	Less than	<i>secsgem.secs.dataitems.ATTRRELN.LESS</i>
3	Less than or equal to	<i>secsgem.secs.dataitems.ATTRRELN.LESS_EQUAL</i>
4	More than	<i>secsgem.secs.dataitems.ATTRRELN.MORE</i>
5	More than or equal to	<i>secsgem.secs.dataitems.ATTRRELN.MORE_EQUAL</i>
6	Value present	<i>secsgem.secs.dataitems.ATTRRELN.PRESENT</i>
7	Value absent	<i>secsgem.secs.dataitems.ATTRRELN.ABSENT</i>
8-63	Error	

Used In Function

- *SecsS14F01*

EQUAL = 0

NOT_EQUAL = 1

LESS = 2

LESS_EQUAL = 3

MORE = 4

MORE_EQUAL = 5

PRESENT = 6

ABSENT = 7

class `secsgem.secs.dataitems.BCEQU` (*value=None*)
 Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarDynamic`

Bin code equivalents.

Types

- `SecsVarString`
- `SecsVarU1`

Used In Function

- `SecsS12F03`
- `SecsS12F04`

class `secsgem.secs.dataitems.BINLT` (*value=None*)
 Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarDynamic`

Bin list.

Types

- `SecsVarString`
- `SecsVarU1`

Used In Function

- `SecsS12F07`
- `SecsS12F09`
- `SecsS12F11`
- `SecsS12F14`
- `SecsS12F16`
- `SecsS12F18`

class `secsgem.secs.dataitems.CEED` (*value=None*)
 Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarBoolean`

Collection event or trace enable/disable code.

Types `SecsVarBoolean`

Length 1

Values

Value	State
True	Enable
False	Disable

Used In Function

- `SecsS02F37`

- SecsS17F05

class secsgem.secs.dataitems.**CEID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Collection event ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F35*
- *SecsS02F37*
- SecsS06F03
- *SecsS06F08*
- SecsS06F09
- *SecsS06F11*
- SecsS06F13
- *SecsS06F15*
- *SecsS06F16*
- SecsS06F17
- SecsS06F18
- SecsS17F05
- SecsS17F09
- SecsS17F10
- SecsS17F11
- SecsS17F12

class secsgem.secs.dataitems.**COLCT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Column count in dies.

Types

- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

class secsgem.secs.dataitems.**COMMACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Establish communications acknowledge.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Accepted	<i>secsgem.secs.dataitems.COMMACK.ACCEPTED</i>
1	Denied, Try Again	<i>secsgem.secs.dataitems.COMMACK.DENIED</i>
2-63	Reserved	

Used In Function

- *SecsS01F14*

ACCEPTED = 0

DENIED = 1

class secsgem.secs.dataitems.**CPACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Command parameter acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
1	Parameter name unknown	<i>secsgem.secs.dataitems.CPACK.PARAMETER_UNKNOWN</i>
2	CPVAL value illegal	<i>secsgem.secs.dataitems.CPACK.CPVAL_ILLEGAL_VALUE</i>
3	CPVAL format illegal	<i>secsgem.secs.dataitems.CPACK.CPVAL_ILLEGAL_FORMAT</i>
4-63	Reserved	

Used In Function

- *SecsS02F42*

PARAMETER_UNKNOWN = 1

CPVAL_ILLEGAL_VALUE = 2

CPVAL_ILLEGAL_FORMAT = 3

class secsgem.secs.dataitems.**CPNAME** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Command parameter name.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F41*
- *SecsS02F42*
- *SecsS02F49*
- *SecsS02F50*
- *SecsS04F21*
- *SecsS04F29*
- *SecsS16F05*
- *SecsS16F27*

class secsgem.secs.dataitems.**CPVAL** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Command parameter name.

Types

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F41*
- *SecsS02F49*
- *SecsS04F21*
- *SecsS04F29*
- *SecsS16F05*
- *SecsS16F27*
- *SecsS18F13*

class `secsgem.secs.dataitems.DATOID` (*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarDynamic`

Data ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F33*
- *SecsS02F35*
- *SecsS02F39*
- *SecsS02F45*
- *SecsS02F49*

- SecsS03F15
- SecsS03F17
- SecsS04F19
- SecsS04F25
- SecsS06F03
- *SecsS06F05*
- *SecsS06F07*
- *SecsS06F08*
- SecsS06F09
- *SecsS06F11*
- SecsS06F13
- *SecsS06F16*
- SecsS06F18
- SecsS06F27
- SecsS13F11
- SecsS13F13
- SecsS13F15
- SecsS14F19
- SecsS14F21
- SecsS14F23
- SecsS15F27
- SecsS15F29
- SecsS15F33
- SecsS15F35
- SecsS15F37
- SecsS15F39
- SecsS15F41
- SecsS15F43
- SecsS15F45
- SecsS15F47
- SecsS15F49
- SecsS16F01
- SecsS16F03
- SecsS16F05
- SecsS16F11
- SecsS16F13

- SecsS17F01
- SecsS17F05
- SecsS17F09

class secsgem.secs.dataitems.**DATALENGTH** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Length of data to be sent.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- SecsS02F39
- SecsS03F15
- SecsS03F29
- SecsS03F31
- SecsS04F25
- *SecsS06F05*
- SecsS13F11
- SecsS14F23
- SecsS16F01
- SecsS16F11
- SecsS18F05
- SecsS18F07
- SecsS19F19

class secsgem.secs.dataitems.**DATLC** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU1*

Data location.

Types *SecsVarU1*

Used In Function

- *SecsS12F19*

class secsgem.secs.dataitems.**DRACK** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Define report acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Acknowledge	<i>secsgem.secs.dataitems.DRACK.ACK</i>
1	Denied, insufficient space	<i>secsgem.secs.dataitems.DRACK.INSUFFICIENT_SPACE</i>
2	Denied, invalid format	<i>secsgem.secs.dataitems.DRACK.INVALID_FORMAT</i>
3	Denied, RPTID already defined	<i>secsgem.secs.dataitems.DRACK.RPTID_REDEFINED</i>
4	Denied, VID doesn't exist	<i>secsgem.secs.dataitems.DRACK.VID_UNKNOWN</i>
5-63	Reserved, other errors	

Used In Function

- *SecsS02F34*

ACK = 0

INSUFFICIENT_SPACE = 1

INVALID_FORMAT = 2

RPTID_REDEFINED = 3

VID_UNKNOWN = 4

class secsgem.secs.dataitems.**DSID** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Data set ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- SecsS06F03
- *SecsS06F08*
- SecsS06F09

class secsgem.secs.dataitems.**DUTMS** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Die units of measure.

Types

- *SecsVarString*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

class secsgem.secs.dataitems.**DVNAME** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Data value name.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- SecsS06F03
- *SecsS06F08*

class secsgem.secs.dataitems.**DVVAL** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Data value.

Types

- *SecsVarArray*

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS06F03*
- *SecsS06F08*
- *SecsS06F09*

class `secsgem.secs.dataitems.EAC` (*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarBinary`

Equipment acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Acknowledge	<code>secsgem.secs.dataitems.EAC.ACK</code>
1	Denied, not all constants exist	<code>secsgem.secs.dataitems.EAC.INVALID_CONSTANT</code>
2	Denied, busy	<code>secsgem.secs.dataitems.EAC.BUSY</code>
3	Denied, constant out of range	<code>secsgem.secs.dataitems.EAC.OUT_OF_RANGE</code>
4-63	Reserved, equipment specific	

Used In Function

- *SecsS02F16*

ACK = 0

INVALID_CONSTANT = 1

BUSY = 2

OUT_OF_RANGE = 3

class secsgem.secs.dataitems.**ECDEF** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Equipment constant default value.

Types

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F30*

class secsgem.secs.dataitems.**ECID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Equipment constant ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F13*
- *SecsS02F15*
- *SecsS02F29*
- *SecsS02F30*

class secsgem.secs.dataitems.**ECMAX**(*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Equipment constant maximum value.

Types

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F30*

class secsgem.secs.dataitems.**ECMIN**(*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Equipment constant minimum value.

Types

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*

- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F30*

class secsgem.secs.dataitems.**ECNAME** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Equipment constant name.

Types

- *SecsVarString*

Used In Function

- *SecsS02F30*

class secsgem.secs.dataitems.**ECV** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Equipment constant value.

Types

- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F14*
- *SecsS02F15*

class secsgem.secs.dataitems.**EDID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Expected data identification.

Types

- *SecsVarBinary*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS09F13*

class secsgem.secs.dataitems.**ERACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Enable/disable event report acknowledge.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Accepted	<i>secsgem.secs.dataitems.ERACK.ACCEPTED</i>
1	Denied, CEID doesn't exist	<i>secsgem.secs.dataitems.ERACK.CEID_UNKNOWN</i>
2-63	Reserved	

Used In Function

- *SecsS02F38*

ACCEPTED = 0

CEID_UNKNOWN = 1

class secsgem.secs.dataitems.**ERRCODE** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Reference point.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*

Used In Function

- *SecsS01F03*
- SecsS01F20
- SecsS03F16
- SecsS03F30
- SecsS03F32
- SecsS04F20
- SecsS04F22
- SecsS04F23
- SecsS04F33
- SecsS04F35
- *SecsS05F14*
- *SecsS05F15*
- *SecsS05F18*
- SecsS13F14
- SecsS13F16
- *SecsS14F02*
- *SecsS14F04*
- SecsS14F06
- SecsS14F08
- SecsS14F10
- SecsS14F12
- SecsS14F14
- SecsS14F16
- SecsS14F18
- SecsS14F26
- SecsS14F28

- SecsS15F18
- SecsS15F20
- SecsS15F22
- SecsS15F24
- SecsS15F26
- SecsS15F28
- SecsS15F30
- SecsS15F32
- SecsS15F34
- SecsS15F36
- SecsS15F38
- SecsS15F40
- SecsS15F42
- SecsS15F44
- SecsS15F48
- SecsS15F53
- SecsS16F12
- SecsS16F14
- SecsS16F16
- SecsS16F18
- SecsS16F24
- SecsS16F26
- SecsS16F28
- SecsS17F02
- SecsS17F04
- SecsS17F06
- SecsS17F08
- SecsS17F10
- SecsS17F12
- SecsS17F14

class secsgem.secs.dataitems.**ERRTEXT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Error description for error code.

Types

- *SecsVarString*

Used In Function

- SecsS01F20
- SecsS03F16
- SecsS03F18
- SecsS03F20
- SecsS03F22
- SecsS03F24
- SecsS03F26
- SecsS03F30
- SecsS03F32
- SecsS04F20
- SecsS04F22
- SecsS04F23
- SecsS04F33
- SecsS04F35
- *SecsS05F14*
- *SecsS05F15*
- *SecsS05F18*
- SecsS13F14
- SecsS13F16
- *SecsS14F02*
- *SecsS14F04*
- SecsS14F06
- SecsS14F08
- SecsS14F10
- SecsS14F12
- SecsS14F14
- SecsS14F16
- SecsS14F18
- SecsS14F26
- SecsS14F28
- SecsS15F28
- SecsS15F30
- SecsS15F32
- SecsS15F34
- SecsS15F36

- SecsS15F38
- SecsS15F40
- SecsS15F42
- SecsS15F44
- SecsS15F48
- SecsS15F53
- SecsS16F12
- SecsS16F14
- SecsS16F16
- SecsS16F18
- SecsS16F24
- SecsS16F26
- SecsS16F28
- SecsS17F04
- SecsS17F08
- SecsS17F14

class secsgem.secs.dataitems.**EXID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Exception identifier.

Types

- *SecsVarString*

Used In Function

- *SecsS05F09*
- *SecsS05F11*
- *SecsS05F13*
- *SecsS05F14*
- *SecsS05F15*
- *SecsS05F17*
- *SecsS05F18*

class secsgem.secs.dataitems.**EXMESSAGE** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Exception message.

Types

- *SecsVarString*

Used In Function

- *SecsS05F09*
- *SecsS05F11*

class secsgem.secs.dataitems.**EXRECVRA** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Exception recovery action.

Types

- *SecsVarString*

Used In Function

- *SecsS05F09*
- *SecsS05F13*

class secsgem.secs.dataitems.**EXTYPE** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Exception type.

Types

- *SecsVarString*

Used In Function

- *SecsS05F09*
- *SecsS05F11*
- *SecsS14F01*
- *SecsS14F02*
- *SecsS14F08*

class secsgem.secs.dataitems.**FFROT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU2*

Film frame rotation.

In degrees from the bottom CW. (Bottom equals zero degrees.) Zero length indicates not used.

Types *SecsVarU2*

Used In Function

- *SecsS12F01*
- *SecsS12F03*

class secsgem.secs.dataitems.**FNLOC** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU2*

Flat/notch location.

In degrees from the bottom CW. (Bottom equals zero degrees.) Zero length indicates not used.

Types *SecsVarU2*

Used In Function

- *SecsS12F01*
- *SecsS12F03*
- *SecsS12F04*

class secsgem.secs.dataitems.**GRANT6** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Permission to send.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Granted	<i>secsgem.secs.dataitems.GRANT6.GRANTED</i>
1	Busy	<i>secsgem.secs.dataitems.GRANT6.BUSY</i>
2	Not interested	<i>secsgem.secs.dataitems.GRANT6.NOT_INTERESTED</i>
3-63	Other error	

Used In Function

- *SecsS06F06*

GRANTED = 0

BUSY = 1

NOT_INTERESTED = 2

class secsgem.secs.dataitems.**GRNT1** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Grant code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Acknowledge	<code>secsgem.secs.dataitems.GRNT1.ACK</code>
1	Busy, try again	<code>secsgem.secs.dataitems.GRNT1.BUSY</code>
2	No space	<code>secsgem.secs.dataitems.GRNT1.NO_SPACE</code>
3	Map too large	<code>secsgem.secs.dataitems.GRNT1.MAP_TOO_LARGE</code>
4	Duplicate ID	<code>secsgem.secs.dataitems.GRNT1.DUPLICATE_ID</code>
5	Material ID not found	<code>secsgem.secs.dataitems.GRNT1.MATERIALID_UNKNOWN</code>
6	Unknown map format	<code>secsgem.secs.dataitems.GRNT1.UNKNOWN_MAP_FORMAT</code>
7-63	Reserved, error	

Used In Function

- `SecsS12F06`

ACK = 0

BUSY = 1

NO_SPACE = 2

MAP_TOO_LARGE = 3

DUPLICATE_ID = 4

MATERIALID_UNKNOWN = 5

UNKNOWN_MAP_FORMAT = 6

class `secsgem.secs.dataitems.HCACK` (*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarBinary`

Host command parameter acknowledge code.

Types `SecsVarBinary`

Length 1

Values

Value	Description	Constant
0	Acknowledge	<code>secsgem.secs.dataitems.HCACK.ACK</code>
1	Denied, invalid command	<code>secsgem.secs.dataitems.HCACK.INVALID_COMMAND</code>
2	Denied, cannot perform now	<code>secsgem.secs.dataitems.HCACK.CANT_PERFORM_NOW</code>
3	Denied, parameter invalid	<code>secsgem.secs.dataitems.HCACK.PARAMETER_INVALID</code>
4	Acknowledge, will finish later	<code>secsgem.secs.dataitems.HCACK.ACK_FINISH_LATER</code>
5	Rejected, already in condition	<code>secsgem.secs.dataitems.HCACK.ALREADY_IN_CONDITION</code>
6	No such object	<code>secsgem.secs.dataitems.HCACK.NO_OBJECT</code>
7-63	Reserved	

Used In Function

- *SecsS02F42*
- *SecsS02F50*

ACK = 0

INVALID_COMMAND = 1

CANT_PERFORM_NOW = 2

PARAMETER_INVALID = 3

ACK_FINISH_LATER = 4

ALREADY_IN_CONDITION = 5

NO_OBJECT = 6

```
class secsgem.secs.dataitems.IDTYP (value=None)
    Bases:      secsgem.secs.dataitems.DataItemBase,      secsgem.secs.variables.
                SecsVarBinary
```

ID type.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Wafer ID	<i>secsgem.secs.dataitems.IDTYP.WAFER</i>
1	Wafer cassette ID	<i>secsgem.secs.dataitems.IDTYP.WAFER_CASSETTE</i>
2	Film frame ID	<i>secsgem.secs.dataitems.IDTYP.FILM_FRAME</i>
3-63	Reserved, error	

Used In Function

- *SecsS12F01*
- *SecsS12F03*
- *SecsS12F04*
- *SecsS12F05*
- *SecsS12F07*
- *SecsS12F09*
- *SecsS12F11*
- *SecsS12F13*
- *SecsS12F14*
- *SecsS12F15*
- *SecsS12F16*
- *SecsS12F17*
- *SecsS12F18*

WAFER = 0

WAFER_CASSETTE = 1

FILM_FRAME = 2

class secsgem.secs.dataitems.**LENGTH** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Service/process program length.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- SecsS02F01
- *SecsS07F01*
- SecsS07F29

class secsgem.secs.dataitems.**LRACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Link report acknowledge code.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Acknowledge	<i>secsgem.secs.dataitems.LRACK.ACK</i>
1	Denied, insufficient space	<i>secsgem.secs.dataitems.LRACK.INSUFFICIENT_SPACE</i>
2	Denied, invalid format	<i>secsgem.secs.dataitems.LRACK.INVALID_FORMAT</i>
3	Denied, CEID already linked	<i>secsgem.secs.dataitems.LRACK.CEID_LINKED</i>
4	Denied, CEID doesn't exist	<i>secsgem.secs.dataitems.LRACK.CEID_UNKNOWN</i>
5	Denied, RPTID doesn't exist	<i>secsgem.secs.dataitems.LRACK.RPTID_UNKNOWN</i>
6-63	Reserved, other errors	

Used In Function

- *SecsS02F36*

ACK = 0

INSUFFICIENT_SPACE = 1

INVALID_FORMAT = 2

CEID_LINKED = 3

CEID_UNKNOWN = 4

RPTID_UNKNOWN = 5

```
class secsgem.secs.dataitems.MAPER(value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Map error.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	ID not found	<i>secsgem.secs.dataitems.MAPER.ID_UNKNOWN</i>
1	Invalid data	<i>secsgem.secs.dataitems.MAPER.INVALID_DATA</i>
2	Format error	<i>secsgem.secs.dataitems.MAPER.FORMAT_ERROR</i>
3-63	Invalid error	

Used In Function

- *SecsS12F19*

ID_UNKNOWN = 0

INVALID_DATA = 1

FORMAT_ERROR = 2

```
class secsgem.secs.dataitems.MAPFT(value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Map data format.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Row format	<i>secsgem.secs.dataitems.MAPFT.ROW</i>
1	Array format	<i>secsgem.secs.dataitems.MAPFT.ARRAY</i>
2	Coordinate format	<i>secsgem.secs.dataitems.MAPFT.COORDINATE</i>
3-63	Error	

Used In Function

- *SecsS12F03*
- *SecsS12F05*

ROW = 0

ARRAY = 1

COORDINATE = 2

```
class secsgem.secs.dataitems.MDACK(value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarBinary
```

Map data acknowledge.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Map received	<i>secsgem.secs.dataitems.MDACK.ACK</i>
1	Format error	<i>secsgem.secs.dataitems.MDACK.FORMAT_ERROR</i>
2	No ID match	<i>secsgem.secs.dataitems.MDACK.UNKNOWN_ID</i>
3	Abort/discard map	<i>secsgem.secs.dataitems.MDACK.ABORT_MAP</i>
4-63	Reserved, error	

Used In Function

- *SecsS12F08*
- *SecsS12F10*
- *SecsS12F12*

ACK = 0

FORMAT_ERROR = 1

UNKNOWN_ID = 2

ABORT_MAP = 3

```
class secsgem.secs.dataitems.MDLN(value=None)
    Bases: secsgem.secs.dataitems.DataItemBase, secsgem.secs.variables.SecsVarString
```

Equipment model type.

Types

- *SecsVarString*

Used In Function

- *SecsS01F02*
- *SecsS01F13*
- *SecsS01F14*

- SecsS07F22
- SecsS07F23
- SecsS07F26
- SecsS07F31
- SecsS07F39
- SecsS07F43

class secsgem.secs.dataitems.**MEXP** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Message expected.

Types

- *SecsVarString*

Used In Function

- *SecsS09F13*

class secsgem.secs.dataitems.**MHEAD** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

SECS message header.

Types *SecsVarBinary*

Length 10

Used In Function

- *SecsS09F01*
- *SecsS09F03*
- *SecsS09F05*
- *SecsS09F07*
- *SecsS09F11*

class secsgem.secs.dataitems.**MID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Material ID.

Types

- *SecsVarBinary*
- *SecsVarString*

Used In Function

- SecsS02F27
- SecsS03F02

- SecsS03F04
- SecsS03F07
- SecsS03F09
- SecsS03F12
- SecsS03F13
- SecsS04F01
- SecsS04F03
- SecsS04F05
- SecsS04F07
- SecsS04F09
- SecsS04F11
- SecsS04F13
- SecsS04F15
- SecsS04F17
- SecsS07F07
- SecsS07F08
- SecsS07F10
- SecsS07F11
- SecsS07F13
- SecsS07F35
- SecsS07F36
- *SecsS12F01*
- *SecsS12F03*
- *SecsS12F04*
- *SecsS12F05*
- *SecsS12F07*
- *SecsS12F09*
- *SecsS12F11*
- *SecsS12F13*
- *SecsS12F14*
- *SecsS12F15*
- *SecsS12F16*
- *SecsS12F17*
- *SecsS12F18*
- SecsS16F11
- SecsS16F13

- SecsS16F15
- SecsS18F10
- SecsS18F11

class secsgem.secs.dataitems.**MLCL** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Message length.

Types

- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS12F04*
- *SecsS12F05*

class secsgem.secs.dataitems.**NULBC** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Column count in dies.

Types

- *SecsVarString*
- *SecsVarU1*

Used In Function

- *SecsS12F01*
- *SecsS12F03*
- *SecsS12F04*

class secsgem.secs.dataitems.**OBJACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU1*

Object acknowledgement code.

Types *SecsVarU1*

Length 1

Values

Value	Description	Constant
0	Successful	<i>secsgem.secs.dataitems.OBJACK.SUCCESSFUL</i>
1	Error	<i>secsgem.secs.dataitems.OBJACK.ERROR</i>
2-63	Reserved	

Used In Function

- *SecsS14F02*
- *SecsS14F04*
- *SecsS14F06*
- *SecsS14F08*
- *SecsS14F10*
- *SecsS14F12*
- *SecsS14F14*
- *SecsS14F16*
- *SecsS14F18*
- *SecsS14F26*
- *SecsS14F28*

SUCCESSFUL = 0

ERROR = 1

class secsgem.secs.dataitems.**OBJID** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Object identifier.

Types

- *SecsVarString*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS01F19*
- *SecsS14F01*
- *SecsS14F02*
- *SecsS14F03*
- *SecsS14F04*

class secsgem.secs.dataitems.**OBJSPEC** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Specific object instance.

Types

- *SecsVarString*

Used In Function

- SecsS02F49
- SecsS13F11
- SecsS13F13
- SecsS13F15
- *SecsS14F01*
- *SecsS14F03*
- SecsS14F05
- SecsS14F07
- SecsS14F09
- SecsS14F10
- SecsS14F11
- SecsS14F13
- SecsS14F15
- SecsS14F16
- SecsS14F17
- SecsS14F19
- SecsS14F25
- SecsS14F27
- SecsS15F43
- SecsS15F47

class secsgem.secs.dataitems.OBJTYPE (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Class of object identifier.

Types

- *SecsVarString*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- SecsS01F19
- *SecsS14F01*
- *SecsS14F03*
- SecsS14F06

- SecsS14F07
- SecsS14F08
- SecsS14F25
- SecsS14F26
- SecsS14F27

class secsgem.secs.dataitems.**OFLACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Acknowledge code for OFFLINE request.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	OFFLINE Acknowledge	<i>secsgem.secs.dataitems.OFLACK.ACK</i>
1-63	Reserved	

Used In Function

- *SecsS01F16*

ACK = 0

class secsgem.secs.dataitems.**ONLACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Acknowledge code for ONLINE request.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	ONLINE Accepted	<i>secsgem.secs.dataitems.ONLACK.ACCEPTED</i>
1	ONLINE Not allowed	<i>secsgem.secs.dataitems.ONLACK.NOT_ALLOWED</i>
2	Already ONLINE	<i>secsgem.secs.dataitems.ONLACK.ALREADY_ON</i>
3-63	Reserved	

Used In Function

- *SecsS01F18*

ACCEPTED = 0

NOT_ALLOWED = 1

ALREADY_ON = 2

class secsgem.secs.dataitems.**ORLOC** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Origin location.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Center die of wafer	<i>secsgem.secs.dataitems.ORLOC.CENTER_DIE</i>
1	Upper right	<i>secsgem.secs.dataitems.ORLOC.UPPER_RIGHT</i>
2	Upper left	<i>secsgem.secs.dataitems.ORLOC.UPPER_LEFT</i>
3	Lower left	<i>secsgem.secs.dataitems.ORLOC.LOWER_LEFT</i>
4	Lower right	<i>secsgem.secs.dataitems.ORLOC.LOWER_RIGHT</i>
5-63	Reserved, error	

Used In Function

- *SecsS12F01*
- *SecsS12F03*
- *SecsS12F04*

CENTER_DIE = 0

UPPER_RIGHT = 1

UPPER_LEFT = 2

LOWER_LEFT = 3

LOWER_RIGHT = 3

class secsgem.secs.dataitems.**PPBODY** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Status variable ID.

Types

- *SecsVarBinary*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS07F03*
- *SecsS07F06*
- *SecsS07F36*
- *SecsS07F37*
- *SecsS07F41*

class secsgem.secs.dataitems.**PPGNT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Process program grant status.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	OK	<i>secsgem.secs.dataitems.PPGNT.OK</i>
1	Already have	<i>secsgem.secs.dataitems.PPGNT.ALREADY_HAVE</i>
2	No space	<i>secsgem.secs.dataitems.PPGNT.NO_SPACE</i>
3	Invalid PPID	<i>secsgem.secs.dataitems.PPGNT.INVALID_PPID</i>
4	Busy, try later	<i>secsgem.secs.dataitems.PPGNT.BUSY</i>
5	Will not accept	<i>secsgem.secs.dataitems.PPGNT.WILL_NOT_ACCEPT</i>
6-63	Reserved, other errors	

Used In Function

- *SecsS07F02*
- *SecsS07F30*

OK = 0

ALREADY_HAVE = 1

NO_SPACE = 2

INVALID_PPID = 3

BUSY = 4

WILL_NOT_ACCEPT = 5

class secsgem.secs.dataitems.**PPID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Process program ID.

Types

- *SecsVarBinary*
- *SecsVarString*

Used In Function

- SecsS02F27
- *SecsS07F01*
- *SecsS07F03*
- *SecsS07F05*
- *SecsS07F06*
- SecsS07F08
- SecsS07F10
- SecsS07F11
- SecsS07F13
- *SecsS07F17*
- *SecsS07F20*
- SecsS07F23
- SecsS07F25
- SecsS07F26
- SecsS07F27
- SecsS07F31
- SecsS07F33
- SecsS07F34
- SecsS07F36
- SecsS07F53

class secsgem.secs.dataitems.**PRAXI** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Process axis.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Rows, top, increasing	<code>secsgem.secs.dataitems.PRAXI.ROWS_TOP_INCR</code>
1	Rows, top, decreasing	<code>secsgem.secs.dataitems.PRAXI.ROWS_TOP_DECR</code>
2	Rows, bottom, increasing	<code>secsgem.secs.dataitems.PRAXI.ROWS_BOT_INCR</code>
3	Rows, bottom, decreasing	<code>secsgem.secs.dataitems.PRAXI.ROWS_BOT_DECR</code>
4	Columns, left, increasing	<code>secsgem.secs.dataitems.PRAXI.COLS_LEFT_INCR</code>
5	Columns, left, decreasing	<code>secsgem.secs.dataitems.PRAXI.COLS_LEFT_DECR</code>
6	Columns, right, increasing	<code>secsgem.secs.dataitems.PRAXI.COLS_RIGHT_INCR</code>
7	Columns, right, decreasing	<code>secsgem.secs.dataitems.PRAXI.COLS_RIGHT_DECR</code>
8-63	Error	

Used In Function

- `SecsS12F01`
- `SecsS12F04`

`ROWS_TOP_INCR = 0`

`ROWS_TOP_DECR = 1`

`ROWS_BOT_INCR = 2`

`ROWS_BOT_DECR = 3`

`COLS_LEFT_INCR = 4`

`COLS_LEFT_DECR = 5`

`COLS_RIGHT_INCR = 6`

`COLS_RIGHT_DECR = 7`

class `secsgem.secs.dataitems.PRDC`*T*(*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarDynamic`

Process die count.

Types

- `SecsVarU8`
- `SecsVarU1`
- `SecsVarU2`
- `SecsVarU4`

Used In Function

- `SecsS12F01`
- `SecsS12F04`

class `secsgem.secs.dataitems.RCMD`(*value=None*)

Bases: `secsgem.secs.dataitems.DataItemBase`, `secsgem.secs.variables.SecsVarDynamic`

Remote command.

Types

- *SecsVarString*
- *SecsVarI1*
- *SecsVarU1*

Used In Function

- SecsS02F21
- *SecsS02F41*
- SecsS02F49

class secsgem.secs.dataitems.**REFP** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Reference point.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*

Used In Function

- *SecsS01F03*
- *SecsS01F11*
- *SecsS01F12*
- SecsS02F23

class secsgem.secs.dataitems.**ROWCT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Row count in dies.

Types

- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

class secsgem.secs.dataitems.**RPSEL** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarU1*

Reference point select.

Types *SecsVarU1*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

class secsgem.secs.dataitems.**RPTID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Report ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F33*
- *SecsS02F35*
- *SecsS06F11*
- *SecsS06F13*
- *SecsS06F16*
- *SecsS06F18*
- *SecsS06F19*
- *SecsS06F21*
- *SecsS06F27*
- *SecsS06F30*
- *SecsS17F01*
- *SecsS17F02*
- *SecsS17F03*
- *SecsS17F04*

- SecsS17F05
- SecsS17F09
- SecsS17F11
- SecsS17F12

class secsgem.secs.dataitems.**RSINF** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Starting location.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*

Used In Function

- *SecsS12F07*
- *SecsS12F14*

class secsgem.secs.dataitems.**SDACK** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Map setup acknowledge.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Received Data	<i>secsgem.secs.dataitems.SDACK.ACK</i>
1-63	Error	

Used In Function

- *SecsS12F02*

ACK = 0

class secsgem.secs.dataitems.**SDBIN** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Send bin information.

Types *SecsVarBinary*

Length 1

Values

Value	Description	Constant
0	Send bin information	<i>secsgem.secs.dataitems.SDBIN.SEND</i>
1	Don't send bin information	<i>secsgem.secs.dataitems.SDBIN.DONT_SEND</i>
2-63	Reserved	

Used In Function

- *SecsS12F17*

SEND = 0

DONT_SEND = 1

class secsgem.secs.dataitems.**SHEAD** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

SECS message header.

Types *SecsVarBinary*

Length 10

Used In Function

- *SecsS09F09*

class secsgem.secs.dataitems.**SOFTREV** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Software revision.

Types

- *SecsVarString*

Used In Function

- *SecsS01F02*
- *SecsS01F13*
- *SecsS01F14*
- *SecsS07F22*
- *SecsS07F23*
- *SecsS07F26*
- *SecsS07F31*
- *SecsS07F39*
- *SecsS07F43*

class secsgem.secs.dataitems.**STRP** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Starting position.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*

Used In Function

- *SecsS12F09*
- *SecsS12F16*

class secsgem.secs.dataitems.**SV** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Status variable value.

Types

- *SecsVarArray*
- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS01F04*
- *SecsS06F01*

class secsgem.secs.dataitems.**SVID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Status variable ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS01F03*
- *SecsS01F11*
- *SecsS01F12*
- *SecsS02F23*

class secsgem.secs.dataitems.**SVNAME** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Status variable name.

Types

- *SecsVarString*

Used In Function

- *SecsS01F12*

class secsgem.secs.dataitems.**TEXT** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Line of characters.

Types

- *SecsVarBinary*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*

- *SecsVarU4*

Used In Function

- *SecsS10F01*
- *SecsS10F03*
- *SecsS10F05*
- *SecsS10F09*

class secsgem.secs.dataitems.**TID** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarBinary*

Terminal ID.

Types *SecsVarBinary*

Length 1

Used In Function

- *SecsS10F01*
- *SecsS10F03*
- *SecsS10F05*
- *SecsS10F07*

class secsgem.secs.dataitems.**TIME** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Time of day.

Types

- *SecsVarString*

Used In Function

- *SecsS02F18*
- *SecsS02F31*

class secsgem.secs.dataitems.**TIMESTAMP** (*value=None*)
 Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Timestamp.

Types

- *SecsVarString*

Used In Function

- *SecsS05F09*
- *SecsS05F11*
- *SecsS05F15*

- SecsS15F41
- SecsS15F44
- SecsS16F05
- SecsS16F07
- SecsS16F09

class secsgem.secs.dataitems.**UNITS** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarString*

Units identifier.

Types

- *SecsVarString*

Used In Function

- *SecsS01F12*
- *SecsS02F30*
- *SecsS02F48*
- *SecsS07F22*

class secsgem.secs.dataitems.**V** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Variable data.

Types

- *SecsVarArray*
- *SecsVarBinary*
- *SecsVarBoolean*
- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarF8*
- *SecsVarF4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS06F11*
- *SecsS06F13*
- *SecsS06F16*
- *SecsS06F20*
- *SecsS06F22*

class secsgem.secs.dataitems.**VID** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Variable ID.

Types

- *SecsVarString*
- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS02F33*
- *SecsS02F45*
- *SecsS02F46*
- *SecsS02F47*
- *SecsS02F48*
- *SecsS06F13*
- *SecsS06F18*
- *SecsS06F22*
- *SecsS17F01*

class secsgem.secs.dataitems.**XDIES** (*value=None*)

Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*

Die size/index X-axis.

Types

- *SecsVarF4*
- *SecsVarF8*

- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

class secsgem.secs.dataitems.**XYPOS** (*value=None*)
Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*
X/Y coordinate position.

Types

- *SecsVarI8*
- *SecsVarI1*
- *SecsVarI2*
- *SecsVarI4*

Used In Function

- *SecsS12F11*
- *SecsS12F18*

class secsgem.secs.dataitems.**YDIES** (*value=None*)
Bases: *secsgem.secs.dataitems.DataItemBase*, *secsgem.secs.variables.SecsVarDynamic*
Die size/index Y-axis.

Types

- *SecsVarF4*
- *SecsVarF8*
- *SecsVarU8*
- *SecsVarU1*
- *SecsVarU2*
- *SecsVarU4*

Used In Function

- *SecsS12F01*
- *SecsS12F04*

3.6.2.3 FunctionBase

Base class for for SECS stream and functions.

class secsgem.secs.functionbase.**StructureDisplayingMeta**

Bases: type

Meta class overriding the default `__repr__` of a class.

mro ()

Return a type's method resolution order.

class secsgem.secs.functionbase.**SecsStreamFunction** (*value=None*)

Bases: object

Secs stream and function base class.

This class is inherited to create a stream/function class. To create a function specific content the class variables `_stream`, `_function` and `_dataFormat` must be overridden.

append (*data*)

Append data to list, if stream/function parameter is a list.

Parameters **data** (*various*) – list item to add

encode ()

Generates the encoded hsms data of the stream/function parameter.

Returns encoded data

Return type string

decode (*data*)

Updates stream/function parameter data from the passed data.

Parameters **data** (*string*) – encoded data

set (*value*)

Updates the value of the stream/function parameter.

Parameters **value** (*various*) – new value for the parameter

get ()

Gets the current value of the stream/function parameter.

Returns current parameter value

Return type various

classmethod **get_format** ()

Gets the format of the function.

Returns returns the string representation of the function

Return type string

3.6.2.4 Functions

Wrappers for SECS stream and functions.

class secsgem.secs.functions.**SecsS00F00** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

Hsms communication.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS00F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS00F00()
S0F0 .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS01F00** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 abort transaction stream 1.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F00()
S1F0 .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS01F01** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 are you online - request.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F01
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F01()
S1F1 W .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS01F02** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 on line data.

Caution: This Stream/function has different structures depending on the source. If it is sent from the equipment side it has the structure below, if it is sent from the host it is an empty list. Be sure to fill the array accordingly.

Structure E->H:

```
{
  MDLN: A[20]
  SOFTREV: A[20]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F02(['secsgem', '0.0.6']) # E->H
S1F2
  <L [2]
    <A "secsgem">
    <A "0.0.6">
  > .
>>> secsgem.SecsS01F02() # H->E
S1F2
  <L> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F03** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

Selected equipment status - request.

Data Items

- *SVID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F03
[
  SVID: U1/U2/U4/U8/I1/I2/I4/I8/A
  ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F03([1, "1337", 12])
S1F3 W
  <L [3]
    <U1 1 >
    <A "1337">
    <U1 12 >
  > .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F04** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 selected equipment status - data.

Data Items

- *SV*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F04
[
    SV: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F04([secsgem.SecsVarU1(1), "text", secsgem.SecsVarU4(1337)])
S1F4
<L [3]
  <U1 1 >
  <A "text">
  <U4 1337 >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F11** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 status variable namelist - request.

Data Items

- *SVID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F11
[
    SVID: U1/U2/U4/U8/I1/I2/I4/I8/A
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F11([1, 1337])
S1F11 W
<L [2]
  <U1 1 >
  <U2 1337 >
> .
```

An empty list will return all available status variables.

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F12** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

status variable namelist - reply.

Data Items

- *SVID*
- *SVNAME*
- *UNITS*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F12
[
  {
    SVID: U1/U2/U4/U8/I1/I2/I4/I8/A
    SVNAME: A
    UNITS: A
  }
  ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F12([{"SVID": 1, "SVNAME": "SV1", "UNITS": "mm"},
... {"SVID": 1337, "SVNAME": "SV2", "UNITS": ""}])
S1F12
<L [2]
  <L [3]
    <U1 1 >
    <A "SV1">
    <A "mm">
  >
  <L [3]
    <U2 1337 >
    <A "SV2">
    <A>
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F13** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

establish communication - request.

Caution: This Stream/function has different structures depending on the source. If it is sent from the equipment side it has the structure below, if it is sent from the host it is an empty list. Be sure to fill the array accordingly.

Structure E->H:

```
{
    MDLN: A[20]
    SOFTREV: A[20]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F13(['secsgem', '0.0.6']) # E->H
S1F13 W
  <L [2]
    <A "secsgem">
    <A "0.0.6">
  > .
>>> secsgem.SecsS01F13() # H->E
S1F13 W
  <L> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F14** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 establish communication - acknowledge.

Caution: This Stream/function has different structures depending on the source. See structure definition below for details. Be sure to fill the array accordingly.

Data Items

- *COMMACK*

Structure E->H:

```
{
    COMMACK: B[1]
    DATA: {
        MDLN: A[20]
        SOFTREV: A[20]
    }
}
```

Structure H->E:

```
{
    COMMACK: B[1]
    DATA: []
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F14({"COMMACK": secsgem.COMMACK.ACCEPTED, "MDLN": ["secsgem",
↪ "0.0.6"]})
S1F14
```

(continues on next page)

(continued from previous page)

```
<L [2]
  <B 0x0>
  <L [2]
    <A "secsgem">
    <A "0.0.6">
  >
> .
```

Parameters **value** (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F15** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
request offline.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F15
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F15()
S1F15 W .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS01F16** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
offline acknowledge.

Data Items

- *OFLACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F16
OFLACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F16(secsgem.OFLACK.ACK)
S1F16
  <B 0x0> .
```

Parameters **value** (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS01F17** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
request online.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F17
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F17()
S1F17 W .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS01F18** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 online acknowledge.

Data Items

- *ONLACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS01F18
ONLACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS01F18(secsgem.ONLACK.ALREADY_ON)
S1F18
<B 0x2> .
```

Parameters **value** (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F00** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 abort transaction stream 2.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F00()
S2F0 .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS02F13** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 equipment constant - request.

Data Items

- *ECID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F13
[
    ECID: U1/U2/U4/U8/I1/I2/I4/I8/A
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F13([1, 1337])
S2F13 W
<L [2]
  <U1 1 >
  <U2 1337 >
> .
```

An empty list will return all available equipment constants.

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F14** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 equipment constant - data.

Data Items

- *ECV*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F14
[
    ECV: L/BOOLEAN/I8/I1/I2/I4/F8/F4/U8/U1/U2/U4/A/B
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F14([secsgem.SecsVarU1(1), "text"])
S2F14
<L [2]
  <U1 1 >
  <A "text">
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F15** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

new equipment constant - send.

Data Items

- *ECID*
- *ECV*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F15
[
  {
    ECID: U1/U2/U4/U8/I1/I2/I4/I8/A
    ECV: L/BOOLEAN/I8/I1/I2/I4/F8/F4/U8/U1/U2/U4/A/B
  }
  ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F15([{"ECID": 1, "ECV": secsgem.SecsVarU4(10)}, {"ECID": "1337", "ECV": "text"}])
↩
S2F15 W
<L [2]
  <L [2]
    <U1 1 >
    <U4 10 >
  >
  <L [2]
    <A "1337">
    <A "text">
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F16** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

new equipment constant - acknowledge.

Data Items

- *EAC*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F16
EAC: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F16(secsgem.EAC.BUSY)
S2F16
<B 0x2> .
```

Parameters **value** (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F17** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 date and time - request.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F17
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F17()
S2F17 W .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS02F18** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 date and time - data.

Data Items

- *TIME*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F18
TIME: A[32]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F18("160816205942")
S2F18
<A "160816205942"> .
```

Parameters **value** (*ASCII string*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F29** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 equipment constant namelist - request.

Data Items

- *ECID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F29
[
    ECID: U1/U2/U4/U8/I1/I2/I4/I8/A
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F29([1, 1337])
S2F29 W
<L [2]
  <U1 1 >
  <U2 1337 >
> .
```

An empty list will return all available equipment constants.

Parameters *value* (*list*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS02F30` (*value=None*)

Bases: `secsgem.secs.functionbase.SecsStreamFunction`

equipment constant namelist.

Data Items

- *ECID*
- *ECNAME*
- *ECMIN*
- *ECMAX*
- *ECDEF*
- *UNITS*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F30
[
    {
        ECID: U1/U2/U4/U8/I1/I2/I4/I8/A
        ECNAME: A
        ECMIN: BOOLEAN/I8/I1/I2/I4/F8/F4/U8/U1/U2/U4/A/B
        ECMAX: BOOLEAN/I8/I1/I2/I4/F8/F4/U8/U1/U2/U4/A/B
        ECDEF: BOOLEAN/I8/I1/I2/I4/F8/F4/U8/U1/U2/U4/A/B
        UNITS: A
    }
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F30([{"ECID": 1, "ECNAME": "EC1", "ECMIN": 0, "ECDEF": 0},
↪secsgem.SecsVarU1(0), "ECMAX": secsgem.SecsVarU1(100), "ECDEF": 0},
↪secsgem.SecsVarU1(50), "UNITS": "mm"}, {"ECID": 1337, "ECNAME": "EC2",
↪", "ECMIN": 0, "ECMAX": 0, "ECDEF": 0, "UNITS": ""}])
S2F30
<L [2]
  <L [6]
    <U1 1 >
    <A "EC1">
    <U1 0 >
    <U1 100 >
    <U1 50 >
    <A "mm">
  >
  <L [6]
    <U2 1337 >
    <A "EC2">
    <A>
    <A>
    <A>
    <A>
  >
> .
```

Parameters *value (list)* – parameters for this function (see example)

class `secsgem.secs.functions.SecsS02F33` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`

define report.

Data Items

- *DATAID*
- *RPTID*
- *VID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F33
{
  DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
  DATA: [
    {
      RPTID: U1/U2/U4/U8/I1/I2/I4/I8/A
      VID: [
        DATA: U1/U2/U4/U8/I1/I2/I4/I8/A
        ...
      ]
    }
    ...
  ]
}
```

Example:

```

>>> import secsgem
>>> secsgem.SecsS02F33({"DATAID": 1, "DATA": [{"RPTID": 1000, "VID": [12, 1337]},
↪ {"RPTID": 1001, "VID": [1, 2355]}]})
S2F33 W
  <L [2]
    <U1 1 >
    <L [2]
      <L [2]
        <U2 1000 >
        <L [2]
          <U1 12 >
          <U2 1337 >
        >
      >
    <L [2]
      <U2 1001 >
      <L [2]
        <U1 1 >
        <U2 2355 >
      >
    >
  >
> .

```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F34** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

define report - acknowledge.

Data Items

- *DRACK*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS02F34
DRACK: B[1]

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS02F34(secsgem.DRACK.INVALID_FORMAT)
S2F34
  <B 0x2> .

```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F35** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

link event report.

Data Items

- *DATAID*

- *CEID*
- *RPTID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F35
{
  DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
  DATA: [
    {
      CEID: U1/U2/U4/U8/I1/I2/I4/I8/A
      RPTID: [
        DATA: U1/U2/U4/U8/I1/I2/I4/I8/A
        ...
      ]
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F35({"DATAID": 1, "DATA": [{"CEID": 1337, "RPTID": [1000, ↵
↵1001]}]})
S2F35 W
<L [2]
  <U1 1 >
  <L [1]
    <L [2]
      <U2 1337 >
      <L [2]
        <U2 1000 >
        <U2 1001 >
      >
    >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F36** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

link event report - acknowledge.

Data Items

- *LRACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F36
LRACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F36(secsgem.LRACK.CEID_UNKNOWN)
S2F36
<B 0x4> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS02F37` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 en-/disable event report.

Data Items

- *CEED*
- *CEID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F37
{
  CEED: BOOLEAN[1]
  CEID: [
    DATA: U1/U2/U4/U8/I1/I2/I4/I8/A
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F37({"CEED": True, "CEID": [1337]})
S2F37 W
<L [2]
  <BOOLEAN True >
  <L [1]
    <U2 1337 >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS02F38` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 en-/disable event report - acknowledge.

Data Items

- *ERACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F38
ERACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F38(secsgem.ERACK.CEID_UNKNOWN)
S2F38
<B 0x1> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F41** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 host command - send.

Data Items

- *RCMD*
- *CPNAME*
- *CPVAL*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F41
{
    RCMD: U1/I1/A
    PARAMS: [
        {
            CPNAME: U1/U2/U4/U8/I1/I2/I4/I8/A
            CPVAL: BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/A/B
        }
        ...
    ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F41({"RCMD": "COMMAND", "PARAMS": [{"CPNAME": "PARAM1", "CPVAL": "VAL1"}, {"CPNAME": "PARAM2", "CPVAL": "VAL2"}]})
S2F41 W
<L [2]
  <A "COMMAND">
  <L [2]
    <L [2]
      <A "PARAM1">
      <A "VAL1">
    >
    <L [2]
      <A "PARAM2">
      <A "VAL2">
    >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS02F42** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

host command - acknowledge.

Data Items

- *HCACK*
- *CPNAME*
- *CPACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS02F42
{
    HCACK: B[1]
    PARAMS: [
        {
            CPNAME: U1/U2/U4/U8/I1/I2/I4/I8/A
            CPACK: B[1]
        }
        ...
    ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS02F42({
    ↪     "PARAMS": [
    ↪ CPVAL_ILLEGAL_VALUE},
    ↪ CPACK.CPVAL_ILLEGAL_FORMAT]})
S2F42
<L [2]
  <B 0x1>
  <L [2]
    <L [2]
      <A "PARAM1">
      <B 0x2>
    >
    <L [2]
      <A "PARAM2">
      <B 0x3>
    >
  >
>
> .
```

Parameters *value (list)* – parameters for this function (see example)

class `secsgem.secs.functions.SecsS05F00` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`

abort transaction stream 5.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F00()
S5F0 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS05F01** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

alarm report - send.

Data Items

- *ALCD*
- *ALID*
- *ALTX*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F01
{
    ALCD: B[1]
    ALID: U1/U2/U4/U8/I1/I2/I4/I8
    ALTX: A[120]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F01({"ALCD": secsgem.ALCD.PERSONAL_SAFETY | secsgem.ALCD.ALARM_
↪SET, "ALID": 100, "ALTX": "text"})
S5F1
<L [3]
  <B 0x81>
  <U1 100 >
  <A "text">
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F02** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

alarm report - acknowledge.

Data Items

- *ACKC5*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F02
ACKC5: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F02(secsgem.ACKC5.ACCEPTED)
S5F2
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS05F03` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`
en-/disable alarm - send.

Data Items

- *ALED*
- *ALID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F03
{
    ALED: B[1]
    ALID: U1/U2/U4/U8/I1/I2/I4/I8
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F03({"ALEID": secsgem.ALED.ENABLE, "ALID": 100})
S5F3
<L [2]
  <B 0x80>
  <U1 100 >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS05F04` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`
en-/disable alarm - acknowledge.

Data Items

- *ACKC5*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F04
ACKC5: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F04(secsgem.ACKC5.ACCEPTED)
```

(continues on next page)

(continued from previous page)

```
S5F4
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F05** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
list alarms - request.

Data Items

- *ALID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F05
[
    ALID: U1/U2/U4/U8/I1/I2/I4/I8
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F05([100, 200])
S5F5 W
<L [2]
    <U1 100 >
    <U1 200 >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F06** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
list alarms - data.

Data Items

- *ALCD*
- *ALID*
- *ALTX*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F06
[
    {
        ALCD: B[1]
        ALID: U1/U2/U4/U8/I1/I2/I4/I8
        ALTX: A[120]
    }
]
```

(continues on next page)

(continued from previous page)

```
...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F06([{"ALCD": secsgem.ALCD.PERSONAL_SAFETY | secsgem.ALCD.
↳ALARM_SET, "ALID": 100, "ALTX": "text"}])
S5F6
<L [1]
  <L [3]
    <B 0x81>
    <U1 100 >
    <A "text">
  >
> .
```

Parameters *value (list)* – parameters for this function (see example)

class `secsgem.secs.functions.SecsS05F07` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 list enabled alarms - request.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F07
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F07()
S5F7 W .
```

Parameters *value (list)* – parameters for this function (see example)

class `secsgem.secs.functions.SecsS05F08` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 list enabled alarms - data.

Data Items

- *ALCD*
- *ALID*
- *ALTX*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F08
[
  {
    ALCD: B[1]
```

(continues on next page)

(continued from previous page)

```

        ALID: U1/U2/U4/U8/I1/I2/I4/I8
        ALTX: A[120]
    }
    ...
]
```

Example:

```

>>> import secsgem
>>> secsgem.SecsS05F08([{"ALCD": secsgem.ALCD.PERSONAL_SAFETY | secsgem.ALCD.
↳ALARM_SET, "ALID": 100, "ALTX": "text"}])
S5F8
<L [1]
  <L [3]
    <B 0x81>
    <U1 100 >
    <A "text">
  >
> .
```

Parameters *value (list)* – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F09** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

exception post - notify.

Data Items

- *TIMESTAMP*
- *EXID*
- *EXTYPE*
- *EXMESSAGE*
- *EXRECVRA*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS05F09
{
    TIMESTAMP: A[32]
    EXID: A[20]
    EXTYPE: A
    EXMESSAGE: A
    EXRECVRA: [
        DATA: A[40]
        ...
    ]
}
```

Example:

```

>>> import secsgem
>>> secsgem.SecsS05F09({
↳"EXID": "EX123",
↳"Exception",
    "TIMESTAMP": "161006221500",
    "EXTYPE": "ALARM",
    "EXMESSAGE":
    "EXRECVRA": ["EXRECVRA1", "EXRECVRA2"] })(continues on next page)
```

(continued from previous page)

```

S5F9
  <L [5]
    <A "161006221500">
    <A "EX123">
    <A "ALARM">
    <A "Exception">
    <L [2]
      <A "EXRECVRA1">
      <A "EXRECVRA2">
    >
  > .

```

Parameters *value (list)* – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F10** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

exception post - confirm.

Structure:

```

>>> import secsgem
>>> secsgem.SecsS05F10
Header only

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS05F10()
S5F10 .

```

Parameters *value (None)* – function has no parameters

class secsgem.secs.functions.**SecsS05F11** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

exception clear - notify.

Data Items

- *TIMESTAMP*
- *EXID*
- *EXTYPE*
- *EXMESSAGE*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS05F11
{
  TIMESTAMP: A[32]
  EXID: A[20]
  EXTYPE: A
  EXMESSAGE: A
}

```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F11({"TIMESTAMP": "161006221500", "EXID": "EX123", "EXTYPE":
↳ "ALARM", "EXMESSAGE": "Exception"})
S5F11
<L [4]
  <A "161006221500">
  <A "EX123">
  <A "ALARM">
  <A "Exception">
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F12** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 exception clear - confirm.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F12
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F12()
S5F12 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS05F13** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 exception recover - request.

Data Items

- *EXID*
- *EXRECVRA*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F13
{
  EXID: A[20]
  EXRECVRA: A[40]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F13({"EXID": "EX123", "EXRECVRA": "EXRECVRA2"})
S5F13 W
```

(continues on next page)

(continued from previous page)

```

<L [2]
  <A "EX123">
  <A "EXRECVRA2">
> .

```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F14** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 exception recover - acknowledge.

Data Items

- *EXID*
- *ACKA*
- *ERRCODE*
- *ERRTEXT*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS05F14
{
  EXID: A[20]
  DATA: {
    ACKA: BOOLEAN[1]
    DATA: {
      ERRCODE: I1/I2/I4/I8
      ERRTEXT: A[120]
    }
  }
}

```

Example::

```

>>> import secsgem
>>> secsgem.SecsS05F14({"EXID": "EX123", "DATA": {"ACKA": False, "DATA": {
↪ "ERRCODE": 10, "ERRTEXT": "Error"}}})
S5F14
<L [2]
  <A "EX123">
  <L [2]
    <BOOLEAN False >
    <L [2]
      <I1 10 >
      <A "Error">
    >
  >
> .

```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F15** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

exception recover complete - notify.

Data Items

- *TIMESTAMP*
- *EXID*
- *ACKA*
- *ERRCODE*
- *ERRTEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F15
{
    TIMESTAMP: A[32]
    EXID: A[20]
    DATA: {
        ACKA: BOOLEAN[1]
        DATA: {
            ERRCODE: I1/I2/I4/I8
            ERRTEXT: A[120]
        }
    }
}
```

Example::

```
>>> import secsgem
>>> secsgem.SecsS05F15({"TIMESTAMP": "161006221500", "EXID": "EX123", "DATA":
↳ {"ACKA": False, "DATA": {"ERRCODE": 10, "ERRTEXT": "Error"}}})
S5F15
<L [3]
  <A "161006221500">
  <A "EX123">
  <L [2]
    <BOOLEAN False >
    <L [2]
      <I1 10 >
      <A "Error">
    >
  >
> .
```

Parameters *value (list)* – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F16** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

exception recover complete - confirm.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F16
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F16()
S5F16 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS05F17** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 exception recover abort - request.

Data Items

- *EXID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F17()
EXID: A[20]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS05F17("EX123")
S5F17 W
<A "EX123"> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS05F18** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 exception recover abort - acknowledge.

Data Items

- *EXID*
- *ACKA*
- *ERRCODE*
- *ERRTEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS05F18()
{
  EXID: A[20]
  DATA: {
    ACKA: BOOLEAN[1]
    DATA: {
      ERRCODE: I1/I2/I4/I8
      ERRTEXT: A[120]
    }
  }
}
```

Example::

```
>>> import secsgem
>>> secsgem.SecsS05F18({"EXID": "EX123", "DATA": {"ACKA": False, "DATA": {
↳ "ERRCODE": 10, "ERRTEXT": "Error"}}})
S5F18
  <L [2]
    <A "EX123">
    <L [2]
      <BOOLEAN False >
      <L [2]
        <I1 10 >
        <A "Error">
      >
    >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F00** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 abort transaction stream 6.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F00()
S6F0 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS06F05** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 multi block data inquiry.

Data Items

- *DATAID*
- *DATALength*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F05
{
  DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
  DATALength: U1/U2/U4/U8/I1/I2/I4/I8
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F05({"DATAID": 1, "DATALENGTH": 1337})
S6F5 W
  <L [2]
    <U1 1 >
    <U2 1337 >
  > .
```

Parameters *value* (*list*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS06F06` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 multi block data grant.

Data Items

- *GRANT6*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F06
GRANT6: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F06(secsgem.GRANT6.BUSY)
S6F6
  <B 0x1> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS06F07` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 data transfer request.

Data Items

- *DATAID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F07
DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F07(1)
S6F7 W
  <U1 1 > .
```

Parameters *value* (*integer*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F08** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 data transfer data.

Data Items

- *DATAID*
- *CEID*
- *DSID*
- *DVNAME*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F08
{
    DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
    CEID: U1/U2/U4/U8/I1/I2/I4/I8/A
    DS: [
        {
            DSID: U1/U2/U4/U8/I1/I2/I4/I8/A
            DV: [
                {
                    DVNAME: U1/U2/U4/U8/I1/I2/I4/I8/A
                    DVVAL: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
                }
                ...
            ]
        }
        ...
    ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F08({
    "DATAID": 1,
    "DS": [{
        "DSID": 1000,
        "DV": [
            {"DVNAME": "VAR1", "DVVAL": "VAR"},
            {"DVNAME": "VAR2", "DVVAL": secsgem.SecsVarU4(100)}
        ]
    }
    "CEID": 1337,
    "DV": [
        {"DVNAME":
S6F8
<L [3]
  <U1 1 >
  <U2 1337 >
  <L [1]
    <L [2]
      <U2 1000 >
      <L [2]
        <L [2]
          <A "VAR1">
          <A "VAR">
        >
      <L [2]
        <A "VAR2">
        <U4 100 >
      >
    ]
  ]
})
```

(continues on next page)

(continued from previous page)

```
>
>
>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS06F11` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 event report.

Data Items

- *DATAID*
- *CEID*
- *RPTID*
- *V*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F11
{
  DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
  CEID: U1/U2/U4/U8/I1/I2/I4/I8/A
  RPT: [
    {
      RPTID: U1/U2/U4/U8/I1/I2/I4/I8/A
      V: [
        DATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
        ...
      ]
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F11({"DATAID": 1, "CEID": 1337, "RPT": [{"RPTID": 1000, "V": [
↪ "VAR", secsgem.SecsVarU4(100)]}]}))
S6F11 W
<L [3]
  <U1 1 >
  <U2 1337 >
  <L [1]
    <L [2]
      <U2 1000 >
      <L [2]
        <A "VAR">
        <U4 100 >
      >
    >
  >
```

(continues on next page)

(continued from previous page)

```
>  
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F12** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
event report - acknowledge.

Data Items

- *ACKC6*

Structure:

```
>>> import secsgem  
>>> secsgem.SecsS06F12  
ACKC6: B[1]
```

Example:

```
>>> import secsgem  
>>> secsgem.SecsS06F12(secsgem.ACKC6.ACCEPTED)  
S6F12  
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F15** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
event report request.

Data Items

- *CEID*

Structure:

```
>>> import secsgem  
>>> secsgem.SecsS06F15  
CEID: U1/U2/U4/U8/I1/I2/I4/I8/A
```

Example:

```
>>> import secsgem  
>>> secsgem.SecsS06F15(1337)  
S6F15 W  
<U2 1337 > .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F16** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
event report data.

Data Items

- *DATAID*
- *CEID*
- *RPTID*
- *V*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F16
{
  DATAID: U1/U2/U4/U8/I1/I2/I4/I8/A
  CEID: U1/U2/U4/U8/I1/I2/I4/I8/A
  RPT: [
    {
      RPTID: U1/U2/U4/U8/I1/I2/I4/I8/A
      V: [
        DATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
        ...
      ]
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F16({"DATAID": 1, "CEID": 1337, "RPT": [{"RPTID": 1000, "V": [
  ↳ "VAR", secsgem.SecsVarU4(100)]}]}))
S6F16
<L [3]
  <U1 1 >
  <U2 1337 >
  <L [1]
    <L [2]
      <U2 1000 >
      <L [2]
        <A "VAR">
        <U4 100 >
      >
    >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS06F19` (*value=None*)

Bases: `secsgem.secs.functionbase.SecsStreamFunction`

individual report request.

Data Items

- *RPTID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F19
RPTID: U1/U2/U4/U8/I1/I2/I4/I8/A
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F19(secsgem.SecsVarU4(1337))
S6F19 W
<U4 1337 > .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F20** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 individual report data.

Data Items

- *V*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F20
[
    V: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F20(["ASD", 1337])
S6F20
<L [2]
  <A "ASD">
  <U2 1337 >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F21** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 annotated individual report request.

Data Items

- *RPTID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F21
RPTID: U1/U2/U4/U8/I1/I2/I4/I8/A
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F21(secsgem.SecsVarU4(1337))
S6F21 W
<U4 1337 > .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS06F22** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 annotated individual report data.

Data Items

- *VID*
- *V*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS06F22
[
  {
    VID: U1/U2/U4/U8/I1/I2/I4/I8/A
    V: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
  }
  ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS06F22([{"VID": "VID1", "V": "ASD"}, {"VID": 2, "V": 1337}])
S6F22
<L [2]
  <L [2]
    <A "VID1">
    <A "ASD">
  >
  <L [2]
    <U1 2 >
    <U2 1337 >
  >
> .
```

Parameters *value* (*list*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F00** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 abort transaction stream 7.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F00()
S7F0 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS07F01** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 process program load - inquire.

Data Items

- *PPID*
- *LENGTH*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F01
{
    PPID: A/B[120]
    LENGTH: U1/U2/U4/U8/I1/I2/I4/I8
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F01({"PPID": "program", "LENGTH": 4})
S7F1 W
<L [2]
  <A "program">
  <U1 4 >
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F02** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 process program load - grant.

Data Items

- *PPGNT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F02
PPGNT: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F02(secsgem.PPGNT.OK)
S7F2
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS07F03` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`
process program - send.

Data Items

- *PPID*
- *PPBODY*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F03
{
    PPID: A/B[120]
    PPBODY: U1/U2/U4/U8/I1/I2/I4/I8/A/B
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F03({"PPID": "program", "PPBODY": secsgem.SecsVarBinary("data
↪")})
S7F3 W
<L [2]
  <A "program">
  <B 0x64 0x61 0x74 0x61>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS07F04` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`
process program - acknowledge.

Data Items

- *ACKC7*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F04
ACKC7: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F04(secsgem.ACKC7.MATRIX_OVERFLOW)
S7F4
<B 0x3> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F05** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 process program - request.

Data Items

- *PPID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F05
PPID: A/B[120]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F05("program")
S7F5 W
<A "program"> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F06** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 process program - data.

Data Items

- *PPID*
- *PPBODY*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F06
{
    PPID: A/B[120]
    PPBODY: U1/U2/U4/U8/I1/I2/I4/I8/A/B
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F06({"PPID": "program", "PPBODY": secsgem.SecsVarBinary("data
↪") })
S7F6
<L [2]
<A "program">
```

(continues on next page)

(continued from previous page)

```
<B 0x64 0x61 0x74 0x61>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F17** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 delete process program - send.

Data Items

- *PPID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F17
[
    PPID: A/B[120]
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F17(["program1", "program2"])
S7F17 W
<L [2]
    <A "program1">
    <A "program2">
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F18** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 delete process program - acknowledge.

Data Items

- *ACKC7*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F18
ACKC7: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F18(secsgem.ACKC7.MODE_UNSUPPORTED)
S7F18
<B 0x5> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F19** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 current equipment process program - request.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F19
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F19()
S7F19 W .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS07F20** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 current equipment process program - data.

Data Items

- *PPID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS07F20
[
    PPID: A/B[120]
    ...
]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS07F20(["program1", "program2"])
S7F20
<L [2]
  <A "program1">
  <A "program2">
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS09F00** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 abort transaction stream 9.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F00
Header only
```


Example:

```
>>> import secsgem
>>> secsgem.SecsS09F00()
S9F0 .
```

Parameters *value* (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS09F01** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 unrecognized device id.

Data Items

- *MHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F01()
MHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F01("HEADERDATA")
S9F1
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS09F03** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 unrecognized stream type.

Data Items

- *MHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F03()
MHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F03("HEADERDATA")
S9F3
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS09F05** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

unrecognized function type.

Data Items

- *MHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F05
MHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F05("HEADERDATA")
S9F5
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS09F07` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

illegal data.

Data Items

- *MHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F07
MHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F07("HEADERDATA")
S9F7
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS09F09` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

transaction timer timeout.

Data Items

- *SHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F09
SHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F09("HEADERDATA")
S9F9
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS09F11` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`

data too long.

Data Items

- *MHEAD*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F11
MHEAD: B[10]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F11("HEADERDATA")
S9F11
<B 0x48 0x45 0x41 0x44 0x45 0x52 0x44 0x41 0x54 0x41> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS09F13` (*value=None*)
Bases: `secsgem.secs.functionbase.SecsStreamFunction`

conversation timeout.

Data Items

- *MEXP*
- *EDID*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS09F13
{
  MEXP: A[6]
  EDID: U1/U2/U4/U8/I1/I2/I4/I8/A/B
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS09F13({"MEXP": "S01E01", "EDID": "data"})
S9F13
<L [2]
  <A "S01E01">
```

(continues on next page)

(continued from previous page)

```
<A "data">
> .
```

Parameters **value** (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS10F00** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
abort transaction stream 10.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS10F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS10F00()
S10F0 .
```

Parameters **value** (*None*) – function has no parameters

class secsgem.secs.functions.**SecsS10F01** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
terminal - request.

Data Items

- *TID*
- *TEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS10F01
{
    TID: B[1]
    TEXT: U1/U2/U4/U8/I1/I2/I4/I8/A/B
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS10F01({"TID": 0, "TEXT": "hello?"})
S10F1
<L [2]
  <B 0x0>
  <A "hello?">
> .
```

Parameters **value** (*dict*) – parameters for this function (see example)

```
class secsgem.secs.functions.SecsS10F02 (value=None)
    Bases: secsgem.secs.functionbase.SecsStreamFunction
    terminal - acknowledge.
```

Data Items

- *ACKC10*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS10F02
ACKC10: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS10F02(secsgem.ACKC10.ACCEPTED)
S10F2
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

```
class secsgem.secs.functions.SecsS10F03 (value=None)
    Bases: secsgem.secs.functionbase.SecsStreamFunction
    terminal single - display.
```

Data Items

- *TID*
- *TEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS10F03
{
    TID: B[1]
    TEXT: U1/U2/U4/U8/I1/I2/I4/I8/A/B
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS10F03({"TID": 0, "TEXT": "hello!"})
S10F3
<L [2]
  <B 0x0>
  <A "hello!">
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

```
class secsgem.secs.functions.SecsS10F04 (value=None)
    Bases: secsgem.secs.functionbase.SecsStreamFunction
    terminal single - acknowledge.
```

Data Items

- *ACKC10*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS10F04
ACKC10: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS10F04(secsgem.ACKC10.TERMINAL_NOT_AVAILABLE)
S10F4
<B 0x2> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F00` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

abort transaction stream 12.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F00()
S12F0 .
```

Parameters *value* (*None*) – function has no parameters

class `secsgem.secs.functions.SecsS12F01` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map setup data - send.

Data Items

- *MID*
- *IDTYP*
- *FNLOC*
- *FFROT*
- *ORLOC*
- *RPSEL*
- *REFP*
- *DUTMS*
- *XDIES*

- *YDIES*
- *ROWCT*
- *COLCT*
- *NULBC*
- *PRDCT*
- *PRAXI*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F01
{
    MID: A/B[80]
    IDTYP: B[1]
    FNLOC: U2
    FFROT: U2
    ORLOC: B
    RPSEL: U1
    REFP: [
        DATA: I1/I2/I4/I8
        ...
    ]
    DUTMS: A
    XDIES: U1/U2/U4/U8/F4/F8
    YDIES: U1/U2/U4/U8/F4/F8
    ROWCT: U1/U2/U4/U8
    COLCT: U1/U2/U4/U8
    NULBC: U1/A
    PRDCT: U1/U2/U4/U8
    PRAXI: B[1]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F01({"MID": "materialID", "IDTYP": secsgem.
↳ IDTYP.WAFER, "FNLOC": 0, "FFROT": 0,
↳ "ORLOC": secsgem.ORLOC.UPPER_LEFT, "RPSEL": 0,
↳ "REFP": [[1,2], [2,3]], "DUTMS": "unit",
↳ "XDIES": 100, "YDIES": 100, "ROWCT": 10,
↳ "COLCT": 10, "NULBC": "{x}", "PRDCT":
↳ 100, "PRAXI": secsgem.PRAXI.ROWS_TOP_INCR, })
S12F1 W
<L [15]
  <A "materialID">
  <B 0x0>
  <U2 0 >
  <U2 0 >
  <B 0x2>
  <U1 0 >
  <L [2]
    <I1 1 2 >
    <I1 2 3 >
  >
  <A "unit">
```

(continues on next page)

(continued from previous page)

```
<U1 100 >
<U1 100 >
<U1 10 >
<U1 10 >
<A "{x}">
<U1 100 >
<B 0x0>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F02** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 map setup data - acknowledge.

Data Items

- *SDACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F02
SDACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F02(secsgem.SDACK.ACK)
S12F2
<B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F03** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 map setup data - request.

Data Items

- *MID*
- *IDTYP*
- *MAPFT*
- *FNLOC*
- *FFROT*
- *ORLOC*
- *PRAXI*
- *BCEQU*
- *NULBC*

Structure:


```
>>> import secsgem
>>> secsgem.SecsS12F03
{
    MID: A/B[80]
    IDTYP: B[1]
    MAPFT: B[1]
    FNLOC: U2
    FFROT: U2
    ORLOC: B
    PRAXI: B[1]
    BCEQU: U1/A
    NULBC: U1/A
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F03({"MID": "materialID", "IDTYP": secsgem.
↪ IDTYP.WAFER_CASSETTE, "MAPFT": secsgem.MAPFT.ROW,
↪ "FNLOC": 0, "FFROT": 0, "ORLOC": secsgem.
↪ ORLOC.LOWER_LEFT, "PRAXI": secsgem.PRAXI.COLS_LEFT_INCR,
↪ "BCEQU": [1, 3, 5, 7], "NULBC": "{x}",
↪ })
S12F3 W
<L [9]
  <A "materialID">
  <B 0x1>
  <B 0x0>
  <U2 0 >
  <U2 0 >
  <B 0x3>
  <B 0x4>
  <U1 1 3 5 7 >
  <A "{x}">
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F04` (*value=None*)

Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map setup data.

Data Items

- *MID*
- *IDTYP*
- *FNLOC*
- *ORLOC*
- *RPSEL*
- *REFP*
- *DUTMS*
- *XDIES*

- *YDIES*
- *ROWCT*
- *COLCT*
- *PRDCT*
- *BCEQU*
- *NULBC*
- *MLCL*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F04
{
  MID: A/B[80]
  IDTYP: B[1]
  FNLOC: U2
  ORLOC: B
  RPSEL: U1
  REFP: [
    DATA: I1/I2/I4/I8
    ...
  ]
  DUTMS: A
  XDIES: U1/U2/U4/U8/F4/F8
  YDIES: U1/U2/U4/U8/F4/F8
  ROWCT: U1/U2/U4/U8
  COLCT: U1/U2/U4/U8
  PRDCT: U1/U2/U4/U8
  BCEQU: U1/A
  NULBC: U1/A
  MLCL: U1/U2/U4/U8
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F04({"MID": "materialID", "IDTYP": secsgem.
↳ IDTYP.FILM_FRAME, "FNLOC": 0, "ORLOC": secsgem.
↳ ORLOC.CENTER_DIE, "RPSEL": 0, "REFP": [[1,2],
↳ [2,3]], "DUTMS": "unit", "XDIES": 100,
↳ "YDIES": 100, "ROWCT": 10, "COLCT": 10,
↳ "PRDCT": 100, "BCEQU": [1, 3, 5, 7],
↳ "NULBC": "{x}", "MLCL": 0, })
S12F4
<L [15]
  <A "materialID">
  <B 0x2>
  <U2 0 >
  <B 0x0>
  <U1 0 >
  <L [2]
    <I1 1 2 >
    <I1 2 3 >
  >
```

(continues on next page)

(continued from previous page)

```

    <A "unit">
    <U1 100 >
    <U1 100 >
    <U1 10 >
    <U1 10 >
    <U1 100 >
    <U1 1 3 5 7 >
    <A "{x}">
    <U1 0 >
> .

```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F05** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 map transmit inquire.

Data Items

- *MID*
- *IDTYP*
- *MAPFT*
- *MLCL*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS12F05
{
    MID: A/B[80]
    IDTYP: B[1]
    MAPFT: B[1]
    MLCL: U1/U2/U4/U8
}

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS12F05({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER, "MAPFT":
↪secsgem.MAPFT.ARRAY, "MLCL": 0})
S12F5 W
<L [4]
  <A "materialID">
  <B 0x0>
  <B 0x1>
  <U1 0 >
> .

```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F06** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*
 map transmit - grant.

Data Items

- *GRNT1*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F06
GRNT1: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F06(secsgem.GRNT1.MATERIALID_UNKNOWN)
S12F6
<B 0x5> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F07` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map data type 1 - send.

Data Items

- *MID*
- *IDTYP*
- *RSINF*
- *BINLT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F07
{
  MID: A/B[80]
  IDTYP: B[1]
  DATA: [
    {
      RSINF: I1/I2/I4/I8[3]
      BINLT: U1/A
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F07({
    "MID": "materialID",
    "IDTYP": "WAFER",
    "DATA": [
        {"RSINF": [1, 2, 3],
         "BINLT": [1, 2, 3, 4]},
        {"RSINF": [4, 5, 6],
         "BINLT": [5, 6, 7, 8]}
    ]})
S12F7 W
<L [3]
<A "materialID">
```

(continues on next page)

(continued from previous page)

```

<B 0x0>
<L [2]
  <L [2]
    <I1 1 2 3 >
    <U1 1 2 3 4 >
  >
  <L [2]
    <I1 4 5 6 >
    <U1 5 6 7 8 >
  >
>
> .

```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F08** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map data type 1 - acknowledge.

Data Items

- *MDACK*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS12F08
MDACK: B[1]

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS12F08(secsgem.MDACK.ABORT_MAP)
S12F8
<B 0x3> .

```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F09** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map data type 2 - send.

Data Items

- *MID*
- *IDTYP*
- *STRP*
- *BINLT*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS12F09
{

```

(continues on next page)

(continued from previous page)

```
MID: A/B[80]
IDTYP: B[1]
STRP: I1/I2/I4/I8[2]
BINLT: U1/A
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F09({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER, "STRP
↪": [0, 1], "BINLT": [1, 2, 3, 4, 5, 6]})
S12F9 W
  <L [4]
    <A "materialID">
    <B 0x0>
    <I1 0 1 >
    <U1 1 2 3 4 5 6 >
  > .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F10** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map data type 2 - acknowledge.

Data Items

- *MDACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F10
MDACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F10(secsgem.MDACK.ACK)
S12F10
  <B 0x0> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F11** (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

map data type 3 - send.

Data Items

- *MID*
- *IDTYP*
- *XYPOS*
- *BINLT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F11
{
  MID: A/B[80]
  IDTYP: B[1]
  DATA: [
    {
      XYPOS: I1/I2/I4/I8[2]
      BINLT: U1/A
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F11({
↪secsgem.IDTYP.WAFER,          "MID": "materialID",          "IDTYP": "WAFER",
↪"BINLT": [1, 2, 3, 4]},      "DATA": [
↪                               {"XYPOS": [1, 2],
↪                               {"XYPOS": [3, 4], "BINLT": [5, 6, 7, 8]}
↪]])
S12F11 W
<L [3]
  <A "materialID">
  <B 0x0>
  <L [2]
    <L [2]
      <I1 1 2 >
      <U1 1 2 3 4 >
    >
    <L [2]
      <I1 3 4 >
      <U1 5 6 7 8 >
    >
  >
>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F12` (*value=None*)

Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map data type 3 - acknowledge.

Data Items

- *MDACK*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F12
MDACK: B[1]
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F12(secsgem.MDACK.FORMAT_ERROR)
S12F12
<B 0x1> .
```

Parameters *value* (*byte*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F13` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 map data type 1 - request.

Data Items

- *MID*
- *IDTYP*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F13
{
    MID: A/B[80]
    IDTYP: B[1]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F13({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER})
S12F13 W
<L [2]
  <A "materialID">
  <B 0x0>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F14` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`
 map data type 1.

Data Items

- *MID*
- *IDTYP*
- *RSINF*
- *BINLT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F14
{
    MID: A/B[80]
```

(continues on next page)

(continued from previous page)

```

IDTYP: B[1]
DATA: [
    {
        RSINF: I1/I2/I4/I8[3]
        BINLT: U1/A
    }
    ...
]
}

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS12F14({
↪secsgem.IDTYP.WAFER,
↪"BINLT": [1, 2, 3, 4]},
↪8]}}))
S12F14
<L [3]
  <A "materialID">
  <B 0x0>
  <L [2]
    <L [2]
      <I1 1 2 3 >
      <U1 1 2 3 4 >
    >
    <L [2]
      <I1 4 5 6 >
      <U1 5 6 7 8 >
    >
  >
>
> .

```

Parameters *value* (*dict*) – parameters for this function (see example)**class** `secsgem.secs.functions.SecsS12F15` (*value=None*)Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map data type 2 - request.

Data Items

- *MID*
- *IDTYP*

Structure:

```

>>> import secsgem
>>> secsgem.SecsS12F15
{
    MID: A/B[80]
    IDTYP: B[1]
}

```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F15({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER})
S12F15 W
  <L [2]
    <A "materialID">
    <B 0x0>
  > .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F16` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map data type 2.

Data Items

- *MID*
- *IDTYP*
- *STRP*
- *BINLT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F16
{
  MID: A/B[80]
  IDTYP: B[1]
  STRP: I1/I2/I4/I8[2]
  BINLT: U1/A
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F16({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER, "STRP
↪": [0, 1], "BINLT": [1, 2, 3, 4, 5, 6]})
S12F16
  <L [4]
    <A "materialID">
    <B 0x0>
    <I1 0 1 >
    <U1 1 2 3 4 5 6 >
  > .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F17` (*value=None*)
 Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map data type 3 - request.

Data Items

- *MID*

- *IDTYP*
- *SDBIN*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F17
{
    MID: A/B[80]
    IDTYP: B[1]
    SDBIN: B[1]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F17({"MID": "materialID", "IDTYP": secsgem.IDTYP.WAFER, "SDBIN": secsgem.SDBIN.DONT_SEND})
↪ S12F17 W
<L [3]
  <A "materialID">
  <B 0x0>
  <B 0x1>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class `secsgem.secs.functions.SecsS12F18` (*value=None*)

Bases: `secsgem.secs.functionbase.SecsStreamFunction`

map data type 3.

Data Items

- *MID*
- *IDTYP*
- *XYPOS*
- *BINLT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F18
{
    MID: A/B[80]
    IDTYP: B[1]
    DATA: [
        {
            XYPOS: I1/I2/I4/I8[2]
            BINLT: U1/A
        }
        ...
    ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F18({ "MID": "materialID",
↪ "IDTYP": secsgem.IDTYP.WAFER, "DATA": [
↪ "XYPOS": [1, 2], "BINLT": [1, 2, 3, 4]}, {"XYPOS": [3, 4],
↪ "BINLT": [5, 6, 7, 8]}])
S12F18
<L [3]
  <A "materialID">
  <B 0x0>
  <L [2]
    <L [2]
      <I1 1 2 >
      <U1 1 2 3 4 >
    >
    <L [2]
      <I1 3 4 >
      <U1 5 6 7 8 >
    >
  >
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS12F19** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
map error report - send.

Data Items

- *MAPER*
- *DATLC*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS12F19
{
  MAPER: B[1]
  DATLC: U1
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS12F19({"MAPER": secsgem.MAPER.INVALID_DATA, "DATLC": 0})
S12F19
<L [2]
  <B 0x1>
  <U1 0 >
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS14F00** (*value=None*)
Bases: *secsgem.secs.functionbase.SecsStreamFunction*
abort transaction stream 14.

Structure:

```
>>> import secsgem
>>> secsgem.SecsS14F00
Header only
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS14F00()
S14F0 .
```

Parameters *value* (*None*) – function has no parameters

class `secsgem.secs.functions.SecsS14F01` (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

GetAttr request.

Data Items

- *OBJSPEC*
- *OBJTYPE*
- *OBJID*
- *ATTRID*
- *ATTRDATA*
- *ATTRRELN*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS14F01
{
  OBJSPEC: A
  OBJTYPE: U1/U2/U4/U8/A
  OBJID: [
    DATA: U1/U2/U4/U8/A
    ...
  ]
  FILTER: [
    {
      ATTRID: U1/U2/U4/U8/A
      ATTRDATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
      ATTRRELN: U1
    }
    ...
  ]
  ATTRID: [
    DATA: U1/U2/U4/U8/A
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS14F01({
    "OBJSPEC": '',
    "OBJTYPE":
    ↳ 'StripMap',
    "OBJID": ['MAP001'],
    "FILTER": [],
    ↳ "ATTRID": ['OriginLocation', 'Rows', 'Columns', 'CellStatus', 'LotID']})
S14F1 W
<L [5]
  <A>
  <A "StripMap">
  <L [1]
    <A "MAP001">
  >
  <L>
  <L [5]
    <A "OriginLocation">
    <A "Rows">
    <A "Columns">
    <A "CellStatus">
    <A "LotID">
  >
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

RELATION = {'EQUAL': 0, 'GREATER': 4, 'GREATEREQUAL': 5, 'LESS': 2, 'LESSEQUAL': 3, 'NONE': 1}

class secsgem.secs.functions.SecsS14F02 (*value=None*)

Bases: *secsgem.secs.functionbase.SecsStreamFunction*

GetAttr data.

Data Items

- *OBJID*
- *ATTRID*
- *ATTRDATA*
- *OBJACK*
- *ERRCODE*
- *ERRTEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS14F02
{
  DATA: [
    {
      OBJID: U1/U2/U4/U8/A
      ATTRIBS: [
        {
          ATTRID: U1/U2/U4/U8/A
          ATTRDATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
        }
        ...
      ]
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    ...
]
ERRORS: {
    OBJACK: U1[1]
    ERROR: [
        {
            ERRCODE: I1/I2/I4/I8
            ERRTEXT: A[120]
        }
        ...
    ]
}
}

```

Example:

```

>>> import secsgem
>>> secsgem.SecsS14F02({
    "DATA": [{
        "OBJID": "MAP001",
        "ATTRIBS": [
            {"ATTRID": "OriginLocation",
             "ATTRDATA": "0"},
            {"ATTRID": "Rows", "ATTRDATA": 4},
            {"ATTRID": "Columns", "ATTRDATA": 4},
            {"ATTRID": "CellStatus", "ATTRDATA": 6},
            {"ATTRID": "LotID", "ATTRDATA": "LOT001"}],
        "ERRORS": {"OBJACK": 0}}]
S14F2
<L [2]
  <L [1]
    <L [2]
      <A "MAP001">
      <L [5]
        <L [2]
          <A "OriginLocation">
          <A "0">
        >
        <L [2]
          <A "Rows">
          <U1 4 >
        >
        <L [2]
          <A "Columns">
          <U1 4 >
        >
        <L [2]
          <A "CellStatus">
          <U1 6 >
        >
        <L [2]
          <A "LotID">
          <A "LOT001">
        >
      >
    >
  >
  <L [2]
    <U1 0 >
    <L>
  >
> .

```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS14F03** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

SetAttr request.

Data Items

- *OBJSPEC*
- *OBJTYPE*
- *OBJID*
- *ATTRID*
- *ATTRDATA*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS14F03
{
  OBJSPEC: A
  OBJTYPE: U1/U2/U4/U8/A
  OBJID: [
    DATA: U1/U2/U4/U8/A
    ...
  ]
  ATTRIBS: [
    {
      ATTRID: U1/U2/U4/U8/A
      ATTRDATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
    }
    ...
  ]
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS14F03({"OBJSPEC": '', "OBJTYPE": 'StripMap', "OBJID": ['MAP001'],
↪ "ATTRIBS": [ {"ATTRID": "CellStatus", "ATTRDATA": "3"} ] })
S14F3 W
<L [4]
  <A>
  <A "StripMap">
  <L [1]
    <A "MAP001">
  >
  <L [1]
    <L [2]
      <A "CellStatus">
      <A "3">
    >
  >
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

class secsgem.secs.functions.**SecsS14F04** (*value=None*)
 Bases: *secsgem.secs.functionbase.SecsStreamFunction*

SetAttr data.

Data Items

- *OBJID*
- *ATTRID*
- *ATTRDATA*
- *OBJACK*
- *ERRCODE*
- *ERRTEXT*

Structure:

```
>>> import secsgem
>>> secsgem.SecsS14F04
{
  DATA: [
    {
      OBJID: U1/U2/U4/U8/A
      ATTRIBS: [
        {
          ATTRID: U1/U2/U4/U8/A
          ATTRDATA: L/BOOLEAN/U1/U2/U4/U8/I1/I2/I4/I8/F4/F8/A/B
        }
        ...
      ]
    }
    ...
  ]
  ERRORS: {
    OBJACK: U1[1]
    ERROR: [
      {
        ERRCODE: I1/I2/I4/I8
        ERRTEXT: A[120]
      }
      ...
    ]
  }
}
```

Example:

```
>>> import secsgem
>>> secsgem.SecsS14F04({
    "DATA": [{
        "OBJID": "MAP001",
        "ATTRIBS": [
            {"ATTRID": "OriginLocation",
             "ATTRDATA": "0"},
            {"ATTRID": "Rows", "ATTRDATA": 4},
            {"ATTRID": "Columns", "ATTRDATA": 4},
            {"ATTRID": "CellStatus", "ATTRDATA": 6},
            {"ATTRID": "LotID", "ATTRDATA": "LOT001"}]
        },
        "ERRORS": {"OBJACK": 0}}])
S14F4
<L [2]
<L [1]
```

(continues on next page)

(continued from previous page)

```
<L [2]
  <A "MAP001">
  <L [5]
    <L [2]
      <A "OriginLocation">
      <A "0">
    >
    <L [2]
      <A "Rows">
      <U1 4 >
    >
    <L [2]
      <A "Columns">
      <U1 4 >
    >
    <L [2]
      <A "CellStatus">
      <U1 6 >
    >
    <L [2]
      <A "LotID">
      <A "LOT001">
    >
  >
>
>
<L [2]
  <U1 0 >
  <L>
>
> .
```

Parameters *value* (*dict*) – parameters for this function (see example)

3.6.2.5 Handler

class `secsgem.secs.handler.SecsHandler` (*address, port, active, session_id, name, custom_connection_handler=None*)

Bases: `secsgem.hsms.handler.HsmsHandler`

Baseclass for creating Host/Equipment models. This layer contains the SECS functionality.

Inherit from this class and override required functions.

register_stream_function (*stream, function, callback*)

Register the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to register callback for
- **function** (*integer*) – function to register callback for
- **callback** (*def callback (connection)*) – method to call when stream and functions is received

unregister_stream_function (*stream, function*)

Unregister the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to unregister callback for
- **function** (*integer*) – function to register callback for

collection_events

Dictionary of available collection events.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.collection_events[123] = {'name': 'collectionEventName', 'dvids': [1, 5]}
```

Key

Id of the collection event (integer)

Data

Dictionary with the following fields

- name** Name of the collection event (string)
- dvids** Data values for the collection event (list of integers)

data_values

Dictionary of available data values.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.data_values[5] = {'name': 'dataValueName', 'ceid': 123 }
```

Key

Id of the data value (integer)

Data

Dictionary with the following fields

- name** Name of the data value (string)
- ceid** Collection event the data value is used for (integer)

alarms

Dictionary of available alarms.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.alarms[137] = {'ceidon': 1371, 'ceidoff': 1372}
```

Key

Id of the alarm (integer)

Data

Dictionary with the following fields

- ceidon** Collection event id for alarm on (integer)
- ceidoff** Collection event id for alarm off (integer)

remote_commands

Dictionary of available remote commands.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.remote_commands["PP_SELECT"] = {'params': [{'name': 'PROGRAM',
↪ 'format': 'A'}], 'ceids': [200, 343]}
```

Key

Name of the remote command (string)

Data

Dictionary with the following fields

params Parameters for the remote command (list of dictionaries)

Parameters

The dictionaries have the following fields

name name of the parameter (string)

format format character of the parameter (string)

ceids Collection events ids the remote command might return (list of integers)

disable_ceids()

Disable all Collection Events.

disable_ceid_reports()

Disable all Collection Event Reports.

list_svs(svs=None)

Get list of available Service Variables.

Returns available Service Variables

Return type list

request_svs(svs)

Request contents of supplied Service Variables.

Parameters **svs** (*list*) – Service Variables to request

Returns values of requested Service Variables

Return type list

request_sv(sv)

Request contents of one Service Variable.

Parameters **sv** (*int*) – id of Service Variable

Returns value of requested Service Variable

Return type various

list_ecs(ecs=None)

Get list of available Equipment Constants.

Returns available Equipment Constants

Return type list

request_ecs (*ecs*)

Request contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – Equipment Constants to request

Returns values of requested Equipment Constants

Return type list

request_ec (*ec*)

Request contents of one Equipment Constant.

Parameters **ec** (*int*) – id of Equipment Constant

Returns value of requested Equipment Constant

Return type various

set_ecs (*ecs*)

Set contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – list containing list of id / value pairs

set_ec (*ec, value*)

Set contents of one Equipment Constant.

Parameters

- **ec** (*int*) – id of Equipment Constant
- **value** (*various*) – new content of Equipment Constant

send_equipment_terminal (*terminal_id, text*)

Set text to equipment terminal.

Parameters

- **terminal_id** (*int*) – ID of terminal
- **text** (*string*) – text to send

get_ceid_name (*ceid*)

Get the name of a collection event.

Parameters **ceid** (*integer*) – ID of collection event

Returns Name of the event or empty string if not found

Return type string

get_dvid_name (*dvid*)

Get the name of a data value.

Parameters **dvid** (*integer*) – ID of data value

Returns Name of the event or empty string if not found

Return type string

are_you_there ()

Check if remote is still replying.

stream_function (*stream, function*)

Get class for stream and function.

Parameters

- **stream** (*int*) – stream to get function for

- **function** (*int*) – function to get

Returns matching stream and function class

Return type secsSxFx class

secs_decode (*packet*)

Get object of decoded stream and function class, or None if no class is available.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – packet to get object for

Returns matching stream and function object

Return type secsSxFx object

callbacks

Property for callback handling.

disable ()

Disables the connection.

enable ()

Enables the connection.

events

Property for event handling.

get_next_system_counter ()

Returns the next System.

Returns System for the next command

Return type integer

on_connection_before_closed (_)

Connection is about to be closed.

on_connection_closed (_)

Connection was closed.

on_connection_established (_)

Connection was established.

on_connection_packet_received (_, *packet*)

Packet received by connection.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – received data packet

send_and_waitfor_response (*packet*)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

Returns Packet that was received

Return type *secsgem.hsms.packets.HsmsPacket*

send_deselect_req ()

Send a Deselect Request to the remote host.

Returns System of the sent request

Return type integer

send_deselect_rsp (*system_id*)

Send a Deselect Response to the remote host.

Parameters `system_id` (*integer*) – System of the request to reply for

send_linktest_req ()

Send a Linktest Request to the remote host.

Returns System of the sent request

Return type integer

send_linktest_rsp (*system_id*)

Send a Linktest Response to the remote host.

Parameters `system_id` (*integer*) – System of the request to reply for

send_reject_rsp (*system_id*, *s_type*, *reason*)

Send a Reject Response to the remote host.

Parameters

- **system_id** (*integer*) – System of the request to reply for
- **s_type** (*integer*) – s_type of rejected message
- **reason** (*integer*) – reason for rejection

send_response (*function*, *system*)

Send response function for system.

Parameters

- **function** (*secsgem.secs.functionbase.SecsStreamFunction*) – function to be sent
- **system** (*integer*) – system to reply to

send_select_req ()

Send a Select Request to the remote host.

Returns System of the sent request

Return type integer

send_select_rsp (*system_id*)

Send a Select Response to the remote host.

Parameters `system_id` (*integer*) – System of the request to reply for

send_separate_req ()

Send a Separate Request to the remote host.

send_stream_function (*packet*)

Send the packet and wait for the response.

Parameters `packet` (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

3.6.3 GEM

3.6.3.1 Handler

class `secsgem.gem.handler.GemHandler` (*address*, *port*, *active*, *session_id*, *name*, *custom_connection_handler=None*)

Bases: *secsgem.secs.handler.SecsHandler*

Baseclass for creating Host/Equipment models. This layer contains GEM functionality.

MDLN = None

model number returned by S01E13/14

SOFTREV = None

software version returned by S01E13/14

enable()

Enables the connection.

disable()

Disables the connection.

on_connection_closed(connection)

Connection was closed.

on_commack_requested()

Get the acknowledgement code for the connection request.

override to accept or deny connection request

Returns 0 when connection is accepted, 1 when connection is denied

Return type integer

send_process_program(ppid, ppbody)

Send a process program.

Parameters

- **ppid** (*string*) – Transferred process programs ID
- **ppbody** (*string*) – Content of process program

request_process_program(ppid)

Request a process program.

Parameters **ppid** (*string*) – Transferred process programs ID

waitfor_communicating(timeout=None)

Wait until connection gets into communicating state. Returns immediately if state is communicating.

Parameters **timeout** (*float*) – seconds to wait before aborting

Returns True if state is communicating, False if timed out

Return type bool

alarms

Dictionary of available alarms.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.alarms[137] = {'ceidon': 1371, 'ceidoff': 1372}
```

Key

Id of the alarm (integer)

Data

Dictionary with the following fields

ceidon Collection event id for alarm on (integer)

ceidoff Collection event id for alarm off (integer)

are_you_there()

Check if remote is still replying.

callbacks

Property for callback handling.

collection_events

Dictionary of available collection events.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.collection_events[123] = {'name': 'collectionEventName', 'dvids':
↳ [1, 5] }
```

Key

Id of the collection event (integer)

Data

Dictionary with the following fields

name Name of the collection event (string)

dvids Data values for the collection event (list of integers)

data_values

Dictionary of available data values.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.data_values[5] = {'name': 'dataValueName', 'ceid': 123 }
```

Key

Id of the data value (integer)

Data

Dictionary with the following fields

name Name of the data value (string)

ceid Collection event the data value is used for (integer)

disable_ceid_reports()

Disable all Collection Event Reports.

disable_ceids()

Disable all Collection Events.

events

Property for event handling.

get_ceid_name(ceid)

Get the name of a collection event.

Parameters **ceid**(integer) – ID of collection event

Returns Name of the event or empty string if not found

Return type string

get_dvid_name (*dvid*)

Get the name of a data value.

Parameters **dvid** (*integer*) – ID of data value

Returns Name of the event or empty string if not found

Return type string

get_next_system_counter ()

Returns the next System.

Returns System for the next command

Return type integer

list_ecs (*ecs=None*)

Get list of available Equipment Constants.

Returns available Equipment Constants

Return type list

list_svs (*svs=None*)

Get list of available Service Variables.

Returns available Service Variables

Return type list

on_connection_before_closed (_)

Connection is about to be closed.

on_connection_established (_)

Connection was established.

on_connection_packet_received (_, *packet*)

Packet received by connection.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – received data packet

register_stream_function (*stream, function, callback*)

Register the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to register callback for
- **function** (*integer*) – function to register callback for
- **callback** (*def callback(connection)*) – method to call when stream and functions is received

remote_commands

Dictionary of available remote commands.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.remote_commands["PP_SELECT"] = {'params': [{'name': 'PROGRAM',
↪ 'format': 'A'}], 'ceids': [200, 343]}
```

Key

Name of the remote command (string)

Data

Dictionary with the following fields

params Parameters for the remote command (list of dictionaries)

Parameters

The dictionaries have the following fields

name name of the parameter (string)

format format character of the parameter (string)

ceids Collection events ids the remote command might return (list of integers)

request_ec (*ec*)

Request contents of one Equipment Constant.

Parameters **ec** (*int*) – id of Equipment Constant

Returns value of requested Equipment Constant

Return type various

request_ecs (*ecs*)

Request contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – Equipment Constants to request

Returns values of requested Equipment Constants

Return type list

request_sv (*sv*)

Request contents of one Service Variable.

Parameters **sv** (*int*) – id of Service Variable

Returns value of requested Service Variable

Return type various

request_svcs (*svs*)

Request contents of supplied Service Variables.

Parameters **svs** (*list*) – Service Variables to request

Returns values of requested Service Variables

Return type list

secs_decode (*packet*)

Get object of decoded stream and function class, or None if no class is available.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – packet to get object for

Returns matching stream and function object

Return type secsSxFx object

send_and_waitfor_response (*packet*)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

Returns Packet that was received

Return type *secsgem.hsms.packets.HsmsPacket*

send_deselect_req()

Send a Deselect Request to the remote host.

Returns System of the sent request

Return type integer

send_deselect_rsp(system_id)

Send a Deselect Response to the remote host.

Parameters **system_id**(*integer*) – System of the request to reply for

send_equipment_terminal(*terminal_id, text*)

Set text to equipment terminal.

Parameters

- **terminal_id**(*int*) – ID of terminal
- **text**(*string*) – text to send

send_linktest_req()

Send a Linktest Request to the remote host.

Returns System of the sent request

Return type integer

send_linktest_rsp(system_id)

Send a Linktest Response to the remote host.

Parameters **system_id**(*integer*) – System of the request to reply for

send_reject_rsp(system_id, s_type, reason)

Send a Reject Response to the remote host.

Parameters

- **system_id**(*integer*) – System of the request to reply for
- **s_type**(*integer*) – s_type of rejected message
- **reason**(*integer*) – reason for rejection

send_response(function, system)

Send response function for system.

Parameters

- **function**(*secsgem.secs.functionbase.SecsStreamFunction*) – function to be sent
- **system**(*integer*) – system to reply to

send_select_req()

Send a Select Request to the remote host.

Returns System of the sent request

Return type integer

send_select_rsp(system_id)

Send a Select Response to the remote host.

Parameters **system_id**(*integer*) – System of the request to reply for

send_separate_req()

Send a Separate Request to the remote host.

send_stream_function (*packet*)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

set_ec (*ec, value*)

Set contents of one Equipment Constant.

Parameters

- **ec** (*int*) – id of Equipment Constant
- **value** (*various*) – new content of Equipment Constant

set_ecs (*ecs*)

Set contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – list containing list of id / value pairs

stream_function (*stream, function*)

Get class for stream and function.

Parameters

- **stream** (*int*) – stream to get function for
- **function** (*int*) – function to get

Returns matching stream and function class

Return type secsSxFx class

unregister_stream_function (*stream, function*)

Unregister the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to unregister callback for
- **function** (*integer*) – function to register callback for

3.6.3.2 HostHandler

class secsgem.gem.hosthandler.**GemHostHandler** (*address, port, active, session_id, name, custom_connection_handler=None*)

Bases: *secsgem.gem.handler.GemHandler*

Baseclass for creating host models. Inherit from this class and override required functions.

clear_collection_events ()

Clear all collection events.

subscribe_collection_event (*ceid, dvs, report_id=None*)

Subscribe to a collection event.

Parameters

- **ceid** (*integer*) – ID of the collection event
- **dvs** (*list of integers*) – DV IDs to add for collection event
- **report_id** (*integer*) – optional - ID for report, autonumbering if None

send_remote_command (*rcmd, params*)

Send a remote command.

Parameters

- **rcmd** (*string*) – Name of command
- **params** (*list of strings*) – DV IDs to add for collection event

delete_process_programs (*ppids*)

Delete a list of process program.

Parameters **ppids** (*list of strings*) – Process programs to delete

get_process_program_list ()

Get process program list.

go_online ()

Set control state to online.

go_offline ()

Set control state to offline.

enable_alarm (*alid*)

Enable alarm.

Parameters **alid** (*secsgem.secs.dataitems.ALID*) – alarm id to enable

disable_alarm (*alid*)

Disable alarm.

Parameters **alid** (*secsgem.secs.dataitems.ALID*) – alarm id to disable

list_alarms (*alids=None*)

List alarms.

Parameters **alids** (*array of int/str*) – alarms to list details for

list_enabled_alarms ()

List enabled alarms.

alarms

Dictionary of available alarms.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.alarms[137] = {'ceidon': 1371, 'ceidoff': 1372}
```

Key

Id of the alarm (integer)

Data

Dictionary with the following fields

ceidon Collection event id for alarm on (integer)

ceidoff Collection event id for alarm off (integer)

are_you_there ()

Check if remote is still replying.

callbacks

Property for callback handling.

collection_events

Dictionary of available collection events.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.collection_events[123] = {'name': 'collectionEventName', 'dvids': [1, 5]}
```

Key

Id of the collection event (integer)

Data

Dictionary with the following fields

name Name of the collection event (string)

dvids Data values for the collection event (list of integers)

data_values

Dictionary of available data values.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, False, 0, "test")
>>> handler.data_values[5] = {'name': 'dataValueName', 'ceid': 123 }
```

Key

Id of the data value (integer)

Data

Dictionary with the following fields

name Name of the data value (string)

ceid Collection event the data value is used for (integer)

disable()

Disables the connection.

disable_ceid_reports()

Disable all Collection Event Reports.

disable_ceids()

Disable all Collection Events.

enable()

Enables the connection.

events

Property for event handling.

get_ceid_name(ceid)

Get the name of a collection event.

Parameters **ceid**(integer) – ID of collection event

Returns Name of the event or empty string if not found

Return type string

get_dvid_name (*dvid*)

Get the name of a data value.

Parameters **dvid** (*integer*) – ID of data value

Returns Name of the event or empty string if not found

Return type string

get_next_system_counter ()

Returns the next System.

Returns System for the next command

Return type integer

list_ecs (*ecs=None*)

Get list of available Equipment Constants.

Returns available Equipment Constants

Return type list

list_svs (*svs=None*)

Get list of available Service Variables.

Returns available Service Variables

Return type list

on_commack_requested ()

Get the acknowledgement code for the connection request.

override to accept or deny connection request

Returns 0 when connection is accepted, 1 when connection is denied

Return type integer

on_connection_before_closed (_)

Connection is about to be closed.

on_connection_closed (*connection*)

Connection was closed.

on_connection_established (_)

Connection was established.

on_connection_packet_received (_, *packet*)

Packet received by connection.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – received data packet

register_stream_function (*stream, function, callback*)

Register the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to register callback for
- **function** (*integer*) – function to register callback for
- **callback** (*def callback (connection)*) – method to call when stream and functions is received

remote_commands

Dictionary of available remote commands.

Example:

```
>>> handler = SecsHandler("127.0.0.1", 5000, True, 0, "test")
>>> handler.remote_commands["PP_SELECT"] = {'params': [{'name': 'PROGRAM',
↪ 'format': 'A'}], 'ceids': [200, 343]}
```

Key

Name of the remote command (string)

Data

Dictionary with the following fields

params Parameters for the remote command (list of dictionaries)

Parameters

The dictionaries have the following fields

name name of the parameter (string)

format format character of the parameter (string)

ceids Collection events ids the remote command might return (list of integers)

request_ec (*ec*)

Request contents of one Equipment Constant.

Parameters **ec** (*int*) – id of Equipment Constant

Returns value of requested Equipment Constant

Return type various

request_ecs (*ecs*)

Request contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – Equipment Constants to request

Returns values of requested Equipment Constants

Return type list

request_process_program (*ppid*)

Request a process program.

Parameters **ppid** (*string*) – Transferred process programs ID

request_sv (*sv*)

Request contents of one Service Variable.

Parameters **sv** (*int*) – id of Service Variable

Returns value of requested Service Variable

Return type various

request_svs (*svs*)

Request contents of supplied Service Variables.

Parameters **svs** (*list*) – Service Variables to request

Returns values of requested Service Variables

Return type list

secs_decode (*packet*)

Get object of decoded stream and function class, or None if no class is available.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – packet to get object for

Returns matching stream and function object

Return type secsSxFx object

send_and_waitfor_response (*packet*)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

Returns Packet that was received

Return type *secsgem.hsms.packets.HsmsPacket*

send_deselect_req ()

Send a Deselect Request to the remote host.

Returns System of the sent request

Return type integer

send_deselect_rsp (*system_id*)

Send a Deselect Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for

send_equipment_terminal (*terminal_id*, *text*)

Set text to equipment terminal.

Parameters

- **terminal_id** (*int*) – ID of terminal
- **text** (*string*) – text to send

send_linktest_req ()

Send a Linktest Request to the remote host.

Returns System of the sent request

Return type integer

send_linktest_rsp (*system_id*)

Send a Linktest Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for

send_process_program (*ppid*, *ppbody*)

Send a process program.

Parameters

- **ppid** (*string*) – Transferred process programs ID
- **ppbody** (*string*) – Content of process program

send_reject_rsp (*system_id*, *s_type*, *reason*)

Send a Reject Response to the remote host.

Parameters

- **system_id** (*integer*) – System of the request to reply for
- **s_type** (*integer*) – s_type of rejected message
- **reason** (*integer*) – reason for rejection

send_response (*function, system*)
Send response function for system.

Parameters

- **function** (*secsgem.secs.functionbase.SecsStreamFunction*) – function to be sent
- **system** (*integer*) – system to reply to

send_select_req ()
Send a Select Request to the remote host.

Returns System of the sent request

Return type integer

send_select_rsp (*system_id*)
Send a Select Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for

send_separate_req ()
Send a Separate Request to the remote host.

send_stream_function (*packet*)
Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

set_ec (*ec, value*)
Set contents of one Equipment Constant.

Parameters

- **ec** (*int*) – id of Equipment Constant
- **value** (*various*) – new content of Equipment Constant

set_ecs (*ecs*)
Set contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – list containing list of id / value pairs

stream_function (*stream, function*)
Get class for stream and function.

Parameters

- **stream** (*int*) – stream to get function for
- **function** (*int*) – function to get

Returns matching stream and function class

Return type secsSxFx class

unregister_stream_function (*stream, function*)
Unregister the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to unregister callback for
- **function** (*integer*) – function to register callback for

waitfor_communicating (*timeout=None*)

Wait until connection gets into communicating state. Returns immediately if state is communicating.

Parameters **timeout** (*float*) – seconds to wait before aborting

Returns True if state is communicating, False if timed out

Return type bool

3.6.3.3 EquipmentHandler

```
class secsgem.gem.equipmenthandler.GemEquipmentHandler (address, port, active,  
session_id, name, cus-  
tom_connection_handler=None,  
ini-  
tial_control_state='ATTEMPT_ONLINE',  
ini-  
tial_online_control_state='REMOTE')
```

Bases: *secsgem.gem.handler.GemHandler*

Baseclass for creating equipment models. Inherit from this class and override required functions.

control_switch_online ()

Operator switches to online control state.

control_switch_offline ()

Operator switches to offline control state.

control_switch_online_local ()

Operator switches to the local online control state.

control_switch_online_remote ()

Operator switches to the local online control state.

data_values

The list of the data values.

Returns Data value list

Return type list of *secsgem.gem.equipmenthandler.DataValue*

on_dv_value_request (*dvid, dv*)

Get the data value depending on its configuration.

Override in inherited class to provide custom data value request handling.

Parameters

- **dvid** (*secsgem.secs.variables.SecsVar*) – Id of the data value encoded in the corresponding type
- **dv** (*secsgem.gem.equipmenthandler.DataValue*) – The data value requested

Returns The value encoded in the corresponding type

Return type *secsgem.secs.variables.SecsVar*

status_variables

The list of the status variables.

Returns Status variable list

Return type list of `secsgem.gem.equipmenthandler.StatusVariables`

on_sv_value_request (*svid, sv*)

Get the status variable value depending on its configuration.

Override in inherited class to provide custom status variable request handling.

Parameters

- **svid** (`secsgem.secs.variables.SecsVar`) – Id of the status variable encoded in the corresponding type
- **sv** (`secsgem.gem.equipmenthandler.StatusVariable`) – The status variable requested

Returns The value encoded in the corresponding type

Return type `secsgem.secs.variables.SecsVar`

collection_events

The list of the collection events.

Returns Collection event list

Return type list of `secsgem.gem.equipmenthandler.CollectionEvent`

registered_reports

The list of the subscribed reports.

Returns Collection event report list

Return type dictionary of subscribed reports

registered_collection_events

The list of the subscribed collection events.

Returns Collection event list

Return type dictionary of `secsgem.gem.equipmenthandler.CollectionEventLink`

trigger_collection_events (*ceids*)

Triggers the supplied collection events.

Parameters **ceids** (*list of various*) – List of collection events

equipment_constants

The list of the equipments constants.

Returns Equipment constant list

Return type list of `secsgem.gem.equipmenthandler.EquipmentConstant`

on_ec_value_request (*ecid, ec*)

Get the equipment constant value depending on its configuration.

Override in inherited class to provide custom equipment constant request handling.

Parameters

- **ecid** (`secsgem.secs.variables.SecsVar`) – Id of the equipment constant encoded in the corresponding type
- **ec** (`secsgem.gem.equipmenthandler.EquipmentConstant`) – The equipment constant requested

Returns The value encoded in the corresponding type

Return type `secsgem.secs.variables.SecsVar`

on_ec_value_update (*ecid, ec, value*)

Set the equipment constant value depending on its configuration.

Override in inherited class to provide custom equipment constant update handling.

Parameters

- **ecid** (`secsgem.secs.variables.SecsVar`) – Id of the equipment constant encoded in the corresponding type
- **ec** (`secsgem.gem.equipmenthandler.EquipmentConstant`) – The equipment constant to be updated
- **value** (`secsgem.secs.variables.SecsVar`) – The value encoded in the corresponding type

alarms

The list of the alarms.

Returns Alarms list

Return type list of `secsgem.gem.equipmenthandler.Alarm`

set_alarm (*alid*)

The list of the alarms.

Parameters **alid** (*str/int*) – Alarm id

clear_alarm (*alid*)

The list of the alarms.

Parameters **alid** (*str/int*) – Alarm id

remote_commands

The list of the remote commands.

Returns Remote command list

Return type list of `secsgem.gem.equipmenthandler.RemoteCommand`

are_you_there ()

Check if remote is still replying.

callbacks

Property for callback handling.

disable ()

Disables the connection.

disable_ceid_reports ()

Disable all Collection Event Reports.

disable_ceids ()

Disable all Collection Events.

enable ()

Enables the connection.

events

Property for event handling.

get_ceid_name (*ceid*)

Get the name of a collection event.

Parameters **ceid** (*integer*) – ID of collection event

Returns Name of the event or empty string if not found

Return type string

get_dvid_name (*dvid*)

Get the name of a data value.

Parameters **dvid** (*integer*) – ID of data value

Returns Name of the event or empty string if not found

Return type string

get_next_system_counter ()

Returns the next System.

Returns System for the next command

Return type integer

list_ecs (*ecs=None*)

Get list of available Equipment Constants.

Returns available Equipment Constants

Return type list

list_svs (*svs=None*)

Get list of available Service Variables.

Returns available Service Variables

Return type list

on_commack_requested ()

Get the acknowledgement code for the connection request.

override to accept or deny connection request

Returns 0 when connection is accepted, 1 when connection is denied

Return type integer

on_connection_before_closed ()

Connection is about to be closed.

on_connection_established ()

Connection was established.

on_connection_packet_received (_, *packet*)

Packet received by connection.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – received data packet

register_stream_function (*stream, function, callback*)

Register the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to register callback for
- **function** (*integer*) – function to register callback for

- **callback** (*def callback (connection)*) – method to call when stream and functions is received

request_ec (*ec*)

Request contents of one Equipment Constant.

Parameters **ec** (*int*) – id of Equipment Constant

Returns value of requested Equipment Constant

Return type various

request_ecs (*ecs*)

Request contents of supplied Equipment Constants.

Parameters **ecs** (*list*) – Equipment Constants to request

Returns values of requested Equipment Constants

Return type list

request_process_program (*ppid*)

Request a process program.

Parameters **ppid** (*string*) – Transferred process programs ID

request_sv (*sv*)

Request contents of one Service Variable.

Parameters **sv** (*int*) – id of Service Variable

Returns value of requested Service Variable

Return type various

request_svs (*svs*)

Request contents of supplied Service Variables.

Parameters **svs** (*list*) – Service Variables to request

Returns values of requested Service Variables

Return type list

secs_decode (*packet*)

Get object of decoded stream and function class, or None if no class is available.

Parameters **packet** (*secsgem.hsms.packets.HsmsPacket*) – packet to get object for

Returns matching stream and function object

Return type secsSxFx object

send_and_waitfor_response (*packet*)

Send the packet and wait for the response.

Parameters **packet** (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

Returns Packet that was received

Return type *secsgem.hsms.packets.HsmsPacket*

send_deselect_req ()

Send a Deselect Request to the remote host.

Returns System of the sent request

Return type integer

send_deselect_rsp (*system_id*)

Send a Deselect Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for

send_equipment_terminal (*terminal_id*, *text*)

Set text to equipment terminal.

Parameters

- **terminal_id** (*int*) – ID of terminal
- **text** (*string*) – text to send

send_linktest_req ()

Send a Linktest Request to the remote host.

Returns System of the sent request

Return type integer

send_linktest_rsp (*system_id*)

Send a Linktest Response to the remote host.

Parameters **system_id** (*integer*) – System of the request to reply for

send_process_program (*ppid*, *ppbody*)

Send a process program.

Parameters

- **ppid** (*string*) – Transferred process programs ID
- **ppbody** (*string*) – Content of process program

send_reject_rsp (*system_id*, *s_type*, *reason*)

Send a Reject Response to the remote host.

Parameters

- **system_id** (*integer*) – System of the request to reply for
- **s_type** (*integer*) – s_type of rejected message
- **reason** (*integer*) – reason for rejection

send_response (*function*, *system*)

Send response function for system.

Parameters

- **function** (*secsgem.secs.functionbase.SecsStreamFunction*) – function to be sent
- **system** (*integer*) – system to reply to

send_select_req ()

Send a Select Request to the remote host.

Returns System of the sent request

Return type integer

send_select_rsp (*system_id*)

Send a Select Response to the remote host.

Parameters `system_id` (*integer*) – System of the request to reply for

send_separate_req ()

Send a Separate Request to the remote host.

send_stream_function (*packet*)

Send the packet and wait for the response.

Parameters `packet` (*secsgem.secs.functionbase.SecsStreamFunction*) – packet to be sent

set_ec (*ec, value*)

Set contents of one Equipment Constant.

Parameters

- **ec** (*int*) – id of Equipment Constant
- **value** (*various*) – new content of Equipment Constant

set_ecs (*ecs*)

Set contents of supplied Equipment Constants.

Parameters `ecs` (*list*) – list containing list of id / value pairs

stream_function (*stream, function*)

Get class for stream and function.

Parameters

- **stream** (*int*) – stream to get function for
- **function** (*int*) – function to get

Returns matching stream and function class

Return type secsSxFx class

unregister_stream_function (*stream, function*)

Unregister the function callback for stream and function.

Parameters

- **stream** (*integer*) – stream to unregister callback for
- **function** (*integer*) – function to register callback for

waitfor_communicating (*timeout=None*)

Wait until connection gets into communicating state. Returns immediately if state is communicating.

Parameters `timeout` (*float*) – seconds to wait before aborting

Returns True if state is communicating, False if timed out

Return type bool

on_connection_closed (*connection*)

Connection was closed.

class secsgem.gem.equipmenthandler.**DataValue** (*dvid, name, value_type, use_callback=True, **kwargs*)

Bases: object

Data value definition.

class secsgem.gem.equipmenthandler.**StatusVariable** (*svid, name, unit, value_type, use_callback=True, **kwargs*)

Bases: object

Status variable definition.

```
class secsgem.gem.equipmenthandler.CollectionEvent (ceid, name, data_values,  
                                                    **kwargs)
```

Bases: object

Collection event definition.

```
class secsgem.gem.equipmenthandler.CollectionEventLink (ce, reports, **kwargs)
```

Bases: object

Representation for registered/linked collection event.

reports

The list of the data values.

Returns List of linked reports

Return type list of *secsgem.gem.equipmenthandler.CollectionEventReport*

```
class secsgem.gem.equipmenthandler.CollectionEventReport (rptid, variables,  
                                                         **kwargs)
```

Bases: object

Report definition for registered collection events.

```
class secsgem.gem.equipmenthandler.EquipmentConstant (ecid, name, min_value,  
                                                         max_value, default_value,  
                                                         unit, value_type,  
                                                         use_callback=True, **kwargs)
```

Bases: object

Equipment constant definition.

3.6.4 Common functionality

Contains helper functions.

- `genindex`
- `modindex`
- `search`

S

`secsgem.common`, [213](#)
`secsgem.secs.dataitems`, [68](#)
`secsgem.secs.functionbase`, [122](#)
`secsgem.secs.functions`, [122](#)
`secsgem.secs.variables`, [40](#)

A

- ABORT_MAP (*secsgem.secs.dataitems.MDACK attribute*), 101
- ABSENT (*secsgem.secs.dataitems.ATTRRELN attribute*), 76
- ACCEPTED (*secsgem.secs.dataitems.ACKC10 attribute*), 71
- ACCEPTED (*secsgem.secs.dataitems.ACKC5 attribute*), 69
- ACCEPTED (*secsgem.secs.dataitems.ACKC6 attribute*), 69
- ACCEPTED (*secsgem.secs.dataitems.ACKC7 attribute*), 70
- ACCEPTED (*secsgem.secs.dataitems.COMMACK attribute*), 79
- ACCEPTED (*secsgem.secs.dataitems.ERACK attribute*), 90
- ACCEPTED (*secsgem.secs.dataitems.ONLACK attribute*), 107
- ACK (*secsgem.secs.dataitems.DRACK attribute*), 84
- ACK (*secsgem.secs.dataitems.EAC attribute*), 86
- ACK (*secsgem.secs.dataitems.GRNT1 attribute*), 97
- ACK (*secsgem.secs.dataitems.HCACK attribute*), 98
- ACK (*secsgem.secs.dataitems.LRACK attribute*), 100
- ACK (*secsgem.secs.dataitems.MDACK attribute*), 101
- ACK (*secsgem.secs.dataitems.OFLACK attribute*), 107
- ACK (*secsgem.secs.dataitems.SDACK attribute*), 114
- ACK_FINISH_LATER (*secsgem.secs.dataitems.HCACK attribute*), 98
- ACKA (*class in secsgem.secs.dataitems*), 71
- ACKC10 (*class in secsgem.secs.dataitems*), 70
- ACKC5 (*class in secsgem.secs.dataitems*), 68
- ACKC6 (*class in secsgem.secs.dataitems*), 69
- ACKC7 (*class in secsgem.secs.dataitems*), 69
- add_peer() (*secsgem.hsms.connectionmanager.HsmsConnectionManager method*), 40
- ALARM_SET (*secsgem.secs.dataitems.ALCD attribute*), 73
- alarms (*secsgem.gem.equipmenthandler.GemEquipmentHandler attribute*), 208
- alarms (*secsgem.gem.handler.GemHandler attribute*), 194
- alarms (*secsgem.gem.hosthandler.GemHostHandler attribute*), 200
- alarms (*secsgem.secs.handler.SecsHandler attribute*), 189
- ALCD (*class in secsgem.secs.dataitems*), 72
- ALED (*class in secsgem.secs.dataitems*), 73
- ALID (*class in secsgem.secs.dataitems*), 73
- ALREADY_HAVE (*secsgem.secs.dataitems.PPGNT attribute*), 109
- ALREADY_IN_CONDITION (*secsgem.secs.dataitems.HCACK attribute*), 98
- ALREADY_ON (*secsgem.secs.dataitems.ONLACK attribute*), 107
- ALTIX (*class in secsgem.secs.dataitems*), 74
- ANYVALUE (*class in secsgem.secs.variables*), 42
- append() (*secsgem.secs.functionbase.SecsStreamFunction method*), 122
- append() (*secsgem.secs.variables.SecsVarArray method*), 46
- are_you_there() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 208
- are_you_there() (*secsgem.gem.handler.GemHandler method*), 194
- are_you_there() (*secsgem.gem.hosthandler.GemHostHandler method*), 200
- are_you_there() (*secsgem.secs.handler.SecsHandler method*), 191
- APPAY (*secsgem.secs.dataitems.MAPFT attribute*), 101
- ATTENTION_FLAGS (*secsgem.secs.dataitems.ALCD attribute*), 73
- ATTRDATA (*class in secsgem.secs.dataitems*), 74
- ATTRID (*class in secsgem.secs.dataitems*), 75
- ATTRRELN (*class in secsgem.secs.dataitems*), 76

B

BCEQU (class in secsgem.secs.dataitems), 76
 BINLT (class in secsgem.secs.dataitems), 77
 BUSY (secsgem.secs.dataitems.EAC attribute), 86
 BUSY (secsgem.secs.dataitems.GRANT6 attribute), 96
 BUSY (secsgem.secs.dataitems.GRNT1 attribute), 97
 BUSY (secsgem.secs.dataitems.PPGNT attribute), 109

C

callbacks (secsgem.gem.equipmenthandler.GemEquipmentHandler attribute), 208
 callbacks (secsgem.gem.handler.GemHandler attribute), 195
 callbacks (secsgem.gem.hosthandler.GemHostHandler attribute), 200
 callbacks (secsgem.hsms.handler.HsmsHandler attribute), 38
 callbacks (secsgem.secs.handler.SecsHandler attribute), 192
 CANT_PERFORM_NOW (secsgem.secs.dataitems.HCACK attribute), 98
 CEED (class in secsgem.secs.dataitems), 77
 CEID (class in secsgem.secs.dataitems), 78
 CEID_LINKED (secsgem.secs.dataitems.LRACK attribute), 100
 CEID_UNKNOWN (secsgem.secs.dataitems.ERACK attribute), 90
 CEID_UNKNOWN (secsgem.secs.dataitems.LRACK attribute), 100
 CENTER_DIE (secsgem.secs.dataitems.ORLOC attribute), 108
 clear_alarm() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
 clear_collection_events() (secsgem.gem.hosthandler.GemHostHandler method), 199
 coding (secsgem.secs.variables.SecsVarJIS8 attribute), 52
 coding (secsgem.secs.variables.SecsVarString attribute), 51
 coding (secsgem.secs.variables.SecsVarText attribute), 49
 COLCT (class in secsgem.secs.dataitems), 78
 collection_events (secsgem.gem.equipmenthandler.GemEquipmentHandler attribute), 207
 collection_events (secsgem.gem.handler.GemHandler attribute), 195
 collection_events (secsgem.gem.hosthandler.GemHostHandler attribute), 200
 collection_events (secsgem.secs.handler.SecsHandler attribute), 189
 CollectionEvent (class in secsgem.gem.equipmenthandler), 213
 CollectionEventLink (class in secsgem.gem.equipmenthandler), 213
 CollectionEventReport (class in secsgem.gem.equipmenthandler), 213
 COLS_LEFT_DECR (secsgem.secs.dataitems.PRAXI attribute), 111
 COLS_LEFT_INCR (secsgem.secs.dataitems.PRAXI attribute), 111
 COLS_RIGHT_DECR (secsgem.secs.dataitems.PRAXI attribute), 111
 COLS_RIGHT_INCR (secsgem.secs.dataitems.PRAXI attribute), 111
 COMMACK (class in secsgem.secs.dataitems), 79
 control_switch_offline() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 206
 control_switch_online() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 206
 control_switch_online_local() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 206
 control_switch_online_remote() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 206
 controlChars (secsgem.secs.variables.SecsVarJIS8 attribute), 52
 controlChars (secsgem.secs.variables.SecsVarString attribute), 51
 controlChars (secsgem.secs.variables.SecsVarText attribute), 49
 COORDINATE (secsgem.secs.dataitems.MAPFT attribute), 101
 CPACK (class in secsgem.secs.dataitems), 79
 CPNAME (class in secsgem.secs.dataitems), 80
 CPVAL (class in secsgem.secs.dataitems), 80
 CPVAL_ILLEGAL_FORMAT (secsgem.secs.dataitems.CPACK attribute), 80
 CPVAL_ILLEGAL_VALUE (secsgem.secs.dataitems.CPACK attribute), 80
 create_connection() (secsgem.hsms.connections.HsmsMultiPassiveServer method), 37

D

DATA_INTEGRITY (secsgem.secs.dataitems.ALCD attribute), 73
 data_values (secsgem.gem.equipmenthandler.GemEquipmentHandler attribute), 206

<code>data_values</code> (<i>secsgem.gem.handler.GemHandler</i> attribute), 195	67
<code>data_values</code> (<i>secsgem.gem.hosthandler.GemHostHandler</i> attribute), 201	63
<code>data_values</code> (<i>secsgem.secs.handler.SecsHandler</i> attribute), 189	
<code>DATAID</code> (class in <i>secsgem.secs.dataitems</i>), 81	<code>decode_item_header()</code> (<i>secsgem.secs.variables.ANYVALUE</i> method), 43
<code>DataItemBase</code> (class in <i>secsgem.secs.dataitems</i>), 68	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVar</i> method), 41
<code>DataItemMeta</code> (class in <i>secsgem.secs.dataitems</i>), 68	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarArray</i> method), 46
<code>DATALength</code> (class in <i>secsgem.secs.dataitems</i>), 83	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarBinary</i> method), 47
<code>DataValue</code> (class in <i>secsgem.gem.equipmenthandler</i>), 212	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarBoolean</i> method), 49
<code>DATLC</code> (class in <i>secsgem.secs.dataitems</i>), 83	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarDynamic</i> method), 42
<code>decode()</code> (<i>secsgem.hsms.packets.HsmsPacket</i> static method), 31	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarF4</i> method), 62
<code>decode()</code> (<i>secsgem.secs.functionbase.SecsStreamFunction</i> method), 122	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarF8</i> method), 60
<code>decode()</code> (<i>secsgem.secs.variables.ANYVALUE</i> method), 43	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarI1</i> method), 56
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarArray</i> method), 46	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarI2</i> method), 58
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarBinary</i> method), 47	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarI4</i> method), 59
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarBoolean</i> method), 49	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarI8</i> method), 55
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarDynamic</i> method), 42	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarJIS8</i> method), 52
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarF4</i> method), 62	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarList</i> method), 45
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarF8</i> method), 60	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarNumber</i> method), 54
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarI1</i> method), 56	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarString</i> method), 51
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarI2</i> method), 58	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarText</i> method), 50
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarI4</i> method), 59	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarU1</i> method), 64
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarI8</i> method), 55	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarU2</i> method), 66
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarJIS8</i> method), 52	<code>decode_item_header()</code> (<i>secsgem.secs.variables.SecsVarU4</i> method),
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarList</i> method), 45	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarNumber</i> method), 54	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarString</i> method), 51	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarText</i> method), 50	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarU1</i> method), 64	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarU2</i> method), 66	
<code>decode()</code> (<i>secsgem.secs.variables.SecsVarU4</i> method),	

66
[decode_item_header\(\)](#) (secsgem.secs.variables.SecsVarU4 method), 67
[decode_item_header\(\)](#) (secsgem.secs.variables.SecsVarU8 method), 63
[delete_process_programs\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 200
DENIED (secsgem.secs.dataitems.COMMACK attribute), 79
DISABLE (secsgem.secs.dataitems.ALED attribute), 73
[disable\(\)](#) (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
[disable\(\)](#) (secsgem.gem.handler.GemHandler method), 194
[disable\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 201
[disable\(\)](#) (secsgem.hsms.connections.HsmsActiveConnection method), 36
[disable\(\)](#) (secsgem.hsms.connections.HsmsMultiPassiveConnection method), 37
[disable\(\)](#) (secsgem.hsms.connections.HsmsPassiveConnection method), 36
[disable\(\)](#) (secsgem.hsms.handler.HsmsHandler method), 38
[disable\(\)](#) (secsgem.secs.handler.SecsHandler method), 192
[disable_alarm\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 200
[disable_ceid_reports\(\)](#) (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
[disable_ceid_reports\(\)](#) (secsgem.gem.handler.GemHandler method), 195
[disable_ceid_reports\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 201
[disable_ceid_reports\(\)](#) (secsgem.secs.handler.SecsHandler method), 190
[disable_ceids\(\)](#) (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
[disable_ceids\(\)](#) (secsgem.gem.handler.GemHandler method), 195
[disable_ceids\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 201
[disable_ceids\(\)](#) (secsgem.gem.secs.handler.SecsHandler method), 190
[disconnect\(\)](#) (secsgem.hsms.connections.HsmsActiveConnection method), 35
[disconnect\(\)](#) (secsgem.hsms.connections.HsmsConnection method), 35
[disconnect\(\)](#) (secsgem.hsms.connections.HsmsMultiPassiveConnection method), 37
[disconnect\(\)](#) (secsgem.hsms.connections.HsmsPassiveConnection method), 37
DONT_SEND (secsgem.secs.dataitems.SDBIN attribute), 115
DRACK (class in secsgem.secs.dataitems), 84
DSID (class in secsgem.secs.dataitems), 84
DUPLICATE_ID (secsgem.secs.dataitems.GRNTI attribute), 97
DUTMS (class in secsgem.secs.dataitems), 85
DVVAL (class in secsgem.secs.dataitems), 85

E

EAC (class in secsgem.secs.dataitems), 86
ECDEF (class in secsgem.secs.dataitems), 87
ECID (class in secsgem.secs.dataitems), 87
ECMAX (class in secsgem.secs.dataitems), 88
ECMIN (class in secsgem.secs.dataitems), 88
ECNAME (class in secsgem.secs.dataitems), 89
ECV (class in secsgem.secs.dataitems), 89
EDID (class in secsgem.secs.dataitems), 90
ENABLE (secsgem.secs.dataitems.ALED attribute), 73
[enable\(\)](#) (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
[enable\(\)](#) (secsgem.gem.handler.GemHandler method), 194
[enable\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 201
[enable\(\)](#) (secsgem.hsms.connections.HsmsActiveConnection method), 36
[enable\(\)](#) (secsgem.hsms.connections.HsmsMultiPassiveConnection method), 37
[enable\(\)](#) (secsgem.hsms.connections.HsmsPassiveConnection method), 36
[enable\(\)](#) (secsgem.hsms.handler.HsmsHandler method), 38
[enable\(\)](#) (secsgem.secs.handler.SecsHandler method), 192
[enable_alarm\(\)](#) (secsgem.gem.hosthandler.GemHostHandler method), 200

```

encode () (secsgem.hsms.packets.HsmsDeselectReqHeader
method), 33
encode () (secsgem.hsms.packets.HsmsDeselectRspHeader
method), 33
encode () (secsgem.hsms.packets.HsmsHeader
method), 31
encode () (secsgem.hsms.packets.HsmsLinktestReqHeader
method), 33
encode () (secsgem.hsms.packets.HsmsLinktestRspHeader
method), 34
encode () (secsgem.hsms.packets.HsmsPacket
method), 31
encode () (secsgem.hsms.packets.HsmsRejectReqHeader
method), 34
encode () (secsgem.hsms.packets.HsmsSelectReqHeader
method), 32
encode () (secsgem.hsms.packets.HsmsSelectRspHeader
method), 32
encode () (secsgem.hsms.packets.HsmsSeparateReqHeader
method), 35
encode () (secsgem.hsms.packets.HsmsStreamFunctionHeader
method), 32
encode () (secsgem.secs.functionbase.SecsStreamFunction
method), 122
encode () (secsgem.secs.variables.ANYVALUE
method), 43
encode () (secsgem.secs.variables.SecsVarArray
method), 46
encode () (secsgem.secs.variables.SecsVarBinary
method), 47
encode () (secsgem.secs.variables.SecsVarBoolean
method), 48
encode () (secsgem.secs.variables.SecsVarDynamic
method), 42
encode () (secsgem.secs.variables.SecsVarF4
method), 62
encode () (secsgem.secs.variables.SecsVarF8
method), 61
encode () (secsgem.secs.variables.SecsVarI1
method), 57
encode () (secsgem.secs.variables.SecsVarI2
method), 58
encode () (secsgem.secs.variables.SecsVarI4
method), 59
encode () (secsgem.secs.variables.SecsVarI8
method), 55
encode () (secsgem.secs.variables.SecsVarJIS8
method), 53
encode () (secsgem.secs.variables.SecsVarList
method), 45
encode () (secsgem.secs.variables.SecsVarNumber
method), 54
encode () (secsgem.secs.variables.SecsVarString
method), 51
encode () (secsgem.secs.variables.SecsVarText
method), 50
encode () (secsgem.secs.variables.SecsVarU1
method), 65
encode () (secsgem.secs.variables.SecsVarU2
method), 66
encode () (secsgem.secs.variables.SecsVarU4
method), 67
encode () (secsgem.secs.variables.SecsVarU8
method), 63
encode_item_header () (secsgem.secs.variables.ANYVALUE
method), 43
encode_item_header () (secsgem.secs.variables.SecsVar
method), 41
encode_item_header () (secsgem.secs.variables.SecsVarArray
method), 46
encode_item_header () (secsgem.secs.variables.SecsVarBinary
method), 48
encode_item_header () (secsgem.secs.variables.SecsVarBoolean
method), 49
encode_item_header () (secsgem.secs.variables.SecsVarDynamic
method), 42
encode_item_header () (secsgem.secs.variables.SecsVarF4
method), 62
encode_item_header () (secsgem.secs.variables.SecsVarF8
method), 61
encode_item_header () (secsgem.secs.variables.SecsVarI1
method), 57
encode_item_header () (secsgem.secs.variables.SecsVarI2
method), 58
encode_item_header () (secsgem.secs.variables.SecsVarI4
method), 59
encode_item_header () (secsgem.secs.variables.SecsVarI8
method), 55
encode_item_header () (secsgem.secs.variables.SecsVarJIS8
method), 53
encode_item_header () (secsgem.secs.variables.SecsVarList
method), 45
encode_item_header () (secsgem.secs.variables.SecsVarNumber
method), 54
encode_item_header () (secsgem.secs.variables.SecsVarString
method), 51
encode_item_header () (secsgem.secs.variables.SecsVarText
method), 50

```

gem.secs.variables.SecsVarText (method), *FORMAT_ERROR* (*secsgem.secs.dataitems.MAPER* attribute), 100
 50
encode_item_header() (*secsgem.secs.dataitems.MDACK* attribute), 101
gem.secs.variables.SecsVarU1 (method),
 65
encode_item_header() (*secsgem.secs.variables.ANYVALUE* attribute), 44
gem.secs.variables.SecsVarU2 (method), *formatCode* (*secsgem.secs.variables.SecsVar* attribute), 40
 66
encode_item_header() (*secsgem.secs.variables.SecsVarArray* attribute), 46
gem.secs.variables.SecsVarU4 (method), *formatCode* (*secsgem.secs.variables.SecsVarBinary* attribute), 47
 67
encode_item_header() (*secsgem.secs.variables.SecsVarBoolean* attribute), 48
gem.secs.variables.SecsVarU8 (method), *formatCode* (*secsgem.secs.variables.SecsVarDynamic* attribute), 42
 63
EQUAL (*secsgem.secs.dataitems.ATTRRELN* attribute), *formatCode* (*secsgem.secs.variables.SecsVarF4* attribute), 62
 76
equipment_constants (*secsgem.secs.variables.SecsVarF8* attribute), 60
gem.gem.equipmenthandler.GemEquipmentHandler (class in *secsgem.gem.equipmenthandler*), 207
EQUIPMENT_SAFETY (*secsgem.secs.dataitems.ALCD* attribute), 72
EQUIPMENT_STATUS_WARNING (*secsgem.secs.dataitems.ALCD* attribute), 73
EquipmentConstant (class in *secsgem.gem.equipmenthandler*), 213
ERACK (class in *secsgem.secs.dataitems*), 90
ERRCODE (class in *secsgem.secs.dataitems*), 90
ERROR (*secsgem.secs.dataitems.ACKC5* attribute), 69
ERROR (*secsgem.secs.dataitems.ACKC6* attribute), 69
ERROR (*secsgem.secs.dataitems.OBJACK* attribute), 105
ERRTEXT (class in *secsgem.secs.dataitems*), 92
events (*secsgem.gem.equipmenthandler.GemEquipmentHandler* attribute), 208
events (*secsgem.gem.handler.GemHandler* attribute), 195
events (*secsgem.gem.hosthandler.GemHostHandler* attribute), 201
events (*secsgem.hsms.connectionmanager.HsmsConnectionManager* attribute), 49
events (*secsgem.hsms.handler.HsmsHandler* attribute), 39
 38
events (*secsgem.secs.handler.SecsHandler* attribute), 192
EXID (class in *secsgem.secs.dataitems*), 94
EXMESSAGE (class in *secsgem.secs.dataitems*), 94
EXRECVRA (class in *secsgem.secs.dataitems*), 95
EXTYPE (class in *secsgem.secs.dataitems*), 95

F

FFROT (class in *secsgem.secs.dataitems*), 95
FILM_FRAME (*secsgem.secs.dataitems.IDTYP* attribute), 99
FNLOC (class in *secsgem.secs.dataitems*), 95

G

GemEquipmentHandler (class in *secsgem.gem.equipmenthandler*), 206
GemHandler (class in *secsgem.gem.handler*), 193
GemHostHandler (class in *secsgem.gem.hosthandler*), 199

`generate()` (`secsgem.secs.variables.ANYVALUE static method`), 44
`generate()` (`secsgem.secs.variables.SecsVar static method`), 40
`generate()` (`secsgem.secs.variables.SecsVarArray static method`), 47
`generate()` (`secsgem.secs.variables.SecsVarBinary static method`), 48
`generate()` (`secsgem.secs.variables.SecsVarBoolean static method`), 49
`generate()` (`secsgem.secs.variables.SecsVarDynamic static method`), 42
`generate()` (`secsgem.secs.variables.SecsVarF4 static method`), 62
`generate()` (`secsgem.secs.variables.SecsVarF8 static method`), 61
`generate()` (`secsgem.secs.variables.SecsVarI1 static method`), 57
`generate()` (`secsgem.secs.variables.SecsVarI2 static method`), 58
`generate()` (`secsgem.secs.variables.SecsVarI4 static method`), 60
`generate()` (`secsgem.secs.variables.SecsVarI8 static method`), 55
`generate()` (`secsgem.secs.variables.SecsVarJIS8 static method`), 53
`generate()` (`secsgem.secs.variables.SecsVarList static method`), 45
`generate()` (`secsgem.secs.variables.SecsVarNumber static method`), 54
`generate()` (`secsgem.secs.variables.SecsVarString static method`), 52
`generate()` (`secsgem.secs.variables.SecsVarText static method`), 50
`generate()` (`secsgem.secs.variables.SecsVarU1 static method`), 65
`generate()` (`secsgem.secs.variables.SecsVarU2 static method`), 66
`generate()` (`secsgem.secs.variables.SecsVarU4 static method`), 68
`generate()` (`secsgem.secs.variables.SecsVarU8 static method`), 64
`get()` (`secsgem.secs.functionbase.SecsStreamFunction method`), 122
`get()` (`secsgem.secs.variables.ANYVALUE method`), 44
`get()` (`secsgem.secs.variables.SecsVarArray method`), 46
`get()` (`secsgem.secs.variables.SecsVarBinary method`), 47
`get()` (`secsgem.secs.variables.SecsVarBoolean method`), 48
`get()` (`secsgem.secs.variables.SecsVarDynamic method`), 41
`get()` (`secsgem.secs.variables.SecsVarF4 method`), 62
`get()` (`secsgem.secs.variables.SecsVarF8 method`), 61
`get()` (`secsgem.secs.variables.SecsVarI1 method`), 57
`get()` (`secsgem.secs.variables.SecsVarI2 method`), 58
`get()` (`secsgem.secs.variables.SecsVarI4 method`), 60
`get()` (`secsgem.secs.variables.SecsVarI8 method`), 56
`get()` (`secsgem.secs.variables.SecsVarJIS8 method`), 53
`get()` (`secsgem.secs.variables.SecsVarList method`), 45
`get()` (`secsgem.secs.variables.SecsVarNumber method`), 54
`get()` (`secsgem.secs.variables.SecsVarString method`), 52
`get()` (`secsgem.secs.variables.SecsVarText method`), 50
`get()` (`secsgem.secs.variables.SecsVarU1 method`), 65
`get()` (`secsgem.secs.variables.SecsVarU2 method`), 66
`get()` (`secsgem.secs.variables.SecsVarU4 method`), 68
`get()` (`secsgem.secs.variables.SecsVarU8 method`), 64
`get_ceid_name()` (`secsgem.gem.equipmenthandler.GemEquipmentHandler method`), 208
`get_ceid_name()` (`secsgem.gem.handler.GemHandler method`), 195
`get_ceid_name()` (`secsgem.gem.hosthandler.GemHostHandler method`), 201
`get_ceid_name()` (`secsgem.secs.handler.SecsHandler method`), 191
`get_connection_id()` (`secsgem.hsms.connectionmanager.HsmsConnectionManager static method`), 40
`get_dvid_name()` (`secsgem.gem.equipmenthandler.GemEquipmentHandler method`), 209
`get_dvid_name()` (`secsgem.gem.handler.GemHandler method`), 195
`get_dvid_name()` (`secsgem.gem.hosthandler.GemHostHandler method`), 201
`get_dvid_name()` (`secsgem.secs.handler.SecsHandler method`), 191
`get_format()` (`secsgem.secs.dataitems.DataItemBase class method`), 68
`get_format()` (`secsgem.secs.functionbase.SecsStreamFunction class method`), 122
`get_format()` (`secsgem.secs.variables.ANYVALUE static method`), 44
`get_format()` (`secsgem.secs.variables.SecsVar static method`), 41
`get_format()` (`secsgem.secs.variables.SecsVarArray`

<i>static method</i>), 46		<i>gem.hsms.handler.HsmsHandler</i> <i>method</i>),	
<code>get_format()</code>	(secs-	38	
<i>gem.secs.variables.SecsVarBinary</i>	<i>static</i>	<code>get_next_system_counter()</code>	(secs-
<i>method</i>), 48		<i>gem.secs.handler.SecsHandler</i>	<i>method</i>),
<code>get_format()</code>	(secs-	192	
<i>gem.secs.variables.SecsVarBoolean</i>	<i>static</i>	<code>get_process_program_list()</code>	(secs-
<i>method</i>), 49		<i>gem.gem.hosthandler.GemHostHandler</i>	
<code>get_format()</code>	(secs-	<i>method</i>), 200	
<i>gem.secs.variables.SecsVarDynamic</i>	<i>static</i>	<code>go_offline()</code>	(secs-
<i>method</i>), 42		<i>gem.gem.hosthandler.GemHostHandler</i>	
<code>get_format()</code>	(secsgem.secs.variables.SecsVarF4	<i>method</i>), 200	
<i>static method</i>), 62		<code>go_online()</code>	(secsgem.gem.hosthandler.GemHostHandler
<code>get_format()</code>	(secsgem.secs.variables.SecsVarF8	<i>method</i>), 200	
<i>static method</i>), 61		GRANT6 (class in secsgem.secs.dataitems), 96	
<code>get_format()</code>	(secsgem.secs.variables.SecsVarI1	GRANTED (secsgem.secs.dataitems.GRANT6 attribute),	
<i>static method</i>), 57		96	
<code>get_format()</code>	(secsgem.secs.variables.SecsVarI2	GRNT1 (class in secsgem.secs.dataitems), 96	
<i>static method</i>), 58			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarI4		
<i>static method</i>), 60			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarI8		
<i>static method</i>), 56			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarJIS8		
<i>static method</i>), 53			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarList		
<i>static method</i>), 44			
<code>get_format()</code>	(secs-		
<i>gem.secs.variables.SecsVarNumber</i>	<i>static</i>		
<i>method</i>), 55			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarString		
<i>static method</i>), 52			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarText		
<i>static method</i>), 51			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarU1		
<i>static method</i>), 65			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarU2		
<i>static method</i>), 67			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarU4		
<i>static method</i>), 68			
<code>get_format()</code>	(secsgem.secs.variables.SecsVarU8		
<i>static method</i>), 64			
<code>get_name_from_format()</code>	(secs-		
<i>gem.secs.variables.SecsVarList</i> <i>static method</i>),			
44			
<code>get_next_system_counter()</code>	(secs-		
<i>gem.gem.equipmenthandler.GemEquipmentHandler</i>	<i>method</i>), 209		
<code>get_next_system_counter()</code>	(secs-		
<i>gem.gem.handler.GemHandler</i>	<i>method</i>),		
196			
<code>get_next_system_counter()</code>	(secs-		
<i>gem.gem.hosthandler.GemHostHandler</i>	<i>method</i>), 202		
<code>get_next_system_counter()</code>	(secs-		

H

<code>has_connection_to()</code>	(secs-		
<i>gem.hsms.connectionmanager.HsmsConnectionManager</i>	<i>method</i>), 40		
HCACK (class in secsgem.secs.dataitems), 97			
HsmsActiveConnection (class in secs-			
<i>gem.hsms.connections</i>), 35			
HsmsConnection (class in secs-			
<i>gem.hsms.connections</i>), 35			
HsmsConnectionManager (class in secs-			
<i>gem.hsms.connectionmanager</i>), 39			
HsmsDeselectReqHeader (class in secs-			
<i>gem.hsms.packets</i>), 33			
HsmsDeselectRspHeader (class in secs-			
<i>gem.hsms.packets</i>), 33			
HsmsHandler (class in secsgem.hsms.handler), 38			
HsmsHeader (class in secsgem.hsms.packets), 31			
HsmsLinktestReqHeader (class in secs-			
<i>gem.hsms.packets</i>), 33			
HsmsLinktestRspHeader (class in secs-			
<i>gem.hsms.packets</i>), 34			
HsmsMultiPassiveConnection (class in secs-			
<i>gem.hsms.connections</i>), 36			
HsmsMultiPassiveServer (class in secs-			
<i>gem.hsms.connections</i>), 37			
HsmsPacket (class in secsgem.hsms.packets), 31			
HsmsPassiveConnection (class in secs-			
<i>gem.hsms.connections</i>), 36			
HsmsRejectReqHeader (class in secs-			
<i>gem.hsms.packets</i>), 34			
HsmsSelectReqHeader (class in secs-			
<i>gem.hsms.packets</i>), 32			
HsmsSelectRspHeader (class in secs-			
<i>gem.hsms.packets</i>), 32			
HsmsSeparateReqHeader (class in secs-			
<i>gem.hsms.packets</i>), 34			

HsmsStreamFunctionHeader (class in secsgem.hsms.packets), 32

I

ID_UNKNOWN (secsgem.secs.dataitems.MAPER attribute), 100

IDTYP (class in secsgem.secs.dataitems), 98

INSUFFICIENT_SPACE (secsgem.secs.dataitems.DRACK attribute), 84

INSUFFICIENT_SPACE (secsgem.secs.dataitems.LRACK attribute), 100

INVALID_COMMAND (secsgem.secs.dataitems.HACK attribute), 98

INVALID_CONSTANT (secsgem.secs.dataitems.EAC attribute), 86

INVALID_DATA (secsgem.secs.dataitems.MAPER attribute), 100

INVALID_FORMAT (secsgem.secs.dataitems.DRACK attribute), 84

INVALID_FORMAT (secsgem.secs.dataitems.LRACK attribute), 100

INVALID_PPID (secsgem.secs.dataitems.PPGNT attribute), 109

IRRECOVERABLE_ERROR (secsgem.secs.dataitems.ALCD attribute), 72

L

LENGTH (class in secsgem.secs.dataitems), 99

LENGTH_ERROR (secsgem.secs.dataitems.ACKC7 attribute), 70

LESS (secsgem.secs.dataitems.ATTRRELN attribute), 76

LESS_EQUAL (secsgem.secs.dataitems.ATTRRELN attribute), 76

list_alarms() (secsgem.gem.hosthandler.GemHostHandler method), 200

list_ecs() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209

list_ecs() (secsgem.gem.handler.GemHandler method), 196

list_ecs() (secsgem.gem.hosthandler.GemHostHandler method), 202

list_ecs() (secsgem.secs.handler.SecsHandler method), 190

list_enabled_alarms() (secsgem.gem.hosthandler.GemHostHandler method), 200

list_svs() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209

list_svs() (secsgem.gem.handler.GemHandler method), 196

list_svs() (secsgem.gem.hosthandler.GemHostHandler method), 202

list_svs() (secsgem.secs.handler.SecsHandler method), 190

LOWER_LEFT (secsgem.secs.dataitems.ORLOC attribute), 108

LOWER_RIGHT (secsgem.secs.dataitems.ORLOC attribute), 108

LRACK (class in secsgem.secs.dataitems), 99

M

MAP_TOO_LARGE (secsgem.secs.dataitems.GRNTI attribute), 97

MAPER (class in secsgem.secs.dataitems), 100

MAPFT (class in secsgem.secs.dataitems), 100

MATERIALID_UNKNOWN (secsgem.secs.dataitems.GRNTI attribute), 97

MATRIX_OVERFLOW (secsgem.secs.dataitems.ACKC7 attribute), 70

MDACK (class in secsgem.secs.dataitems), 101

MDLN (class in secsgem.secs.dataitems), 101

MDLN (secsgem.gem.handler.GemHandler attribute), 193

MEXP (class in secsgem.secs.dataitems), 102

MHEAD (class in secsgem.secs.dataitems), 102

MID (class in secsgem.secs.dataitems), 102

MLCL (class in secsgem.secs.dataitems), 104

MODE_UNSUPPORTED (secsgem.secs.dataitems.ACKC7 attribute), 70

MORE (secsgem.secs.dataitems.ATTRRELN attribute), 76

MORE_EQUAL (secsgem.secs.dataitems.ATTRRELN attribute), 76

mro() (secsgem.secs.functionbase.StructureDisplayingMeta method), 122

N

NO_OBJECT (secsgem.secs.dataitems.HACK attribute), 98

NO_PERMISSION (secsgem.secs.dataitems.ACKC7 attribute), 70

NO_SPACE (secsgem.secs.dataitems.GRNTI attribute), 97

NO_SPACE (secsgem.secs.dataitems.PPGNT attribute), 109

NOT_ALLOWED (secsgem.secs.dataitems.ONLACK attribute), 107

NOT_DISPLAYED (secsgem.secs.dataitems.ACKC10 attribute), 71

NOT_EQUAL (secsgem.secs.dataitems.ATTRRELN attribute), 76

NOT_INTERESTED (secsgem.secs.dataitems.GRANT6 attribute), 96

NULBC (class in secsgem.secs.dataitems), 104

O

OBJACK (class in secsgem.secs.dataitems), 104

OBJID (class in secsgem.secs.dataitems), 105

OBJSPEC (class in secsgem.secs.dataitems), 105
 OBJTYPE (class in secsgem.secs.dataitems), 106
 OFLACK (class in secsgem.secs.dataitems), 107
 OK (secsgem.secs.dataitems.PPGNT attribute), 109
 on_commack_requested() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209
 on_commack_requested() (secsgem.gem.handler.GemHandler method), 194
 on_commack_requested() (secsgem.gem.hosthandler.GemHostHandler method), 202
 on_connected() (secsgem.gem.hsms.connections.HsmsMultiPassiveConnection method), 36
 on_connection_before_closed() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209
 on_connection_before_closed() (secsgem.gem.handler.GemHandler method), 196
 on_connection_before_closed() (secsgem.gem.hosthandler.GemHostHandler method), 202
 on_connection_before_closed() (secsgem.gem.hsms.handler.HsmsHandler method), 38
 on_connection_before_closed() (secsgem.secs.handler.SecsHandler method), 192
 on_connection_closed() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 212
 on_connection_closed() (secsgem.gem.handler.GemHandler method), 194
 on_connection_closed() (secsgem.gem.hosthandler.GemHostHandler method), 202
 on_connection_closed() (secsgem.gem.hsms.handler.HsmsHandler method), 38
 on_connection_closed() (secsgem.secs.handler.SecsHandler method), 192
 on_connection_established() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209
 on_connection_established() (secsgem.gem.handler.GemHandler method), 196
 on_connection_established() (secsgem.gem.hosthandler.GemHostHandler method), 202
 on_connection_established() (secsgem.gem.hsms.handler.HsmsHandler method), 38
 on_connection_established() (secsgem.secs.handler.SecsHandler method), 192
 on_connection_packet_received() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 209
 on_connection_packet_received() (secsgem.gem.handler.GemHandler method), 196
 on_connection_packet_received() (secsgem.gem.hosthandler.GemHostHandler method), 202
 on_connection_packet_received() (secsgem.gem.hsms.handler.HsmsHandler method), 38
 on_connection_packet_received() (secsgem.secs.handler.SecsHandler method), 192
 on_dv_value_request() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 206
 on_ec_value_request() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 207
 on_ec_value_update() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
 on_sv_value_request() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 207
 ONLACK (class in secsgem.secs.dataitems), 107
 ORLOC (class in secsgem.secs.dataitems), 107
 OUT_OF_RANGE (secsgem.secs.dataitems.EAC attribute), 87

P

PARAMETER_CONTROL_ERROR (secsgem.secs.dataitems.ALCD attribute), 72
 PARAMETER_CONTROL_WARNING (secsgem.secs.dataitems.ALCD attribute), 72
 PARAMETER_INVALID (secsgem.secs.dataitems.HCACK attribute), 98
 PARAMETER_UNKNOWN (secsgem.secs.dataitems.CPACK attribute), 80
 PERFORMED_LATER (secsgem.secs.dataitems.ACKC7 attribute), 70
 PERSONAL_SAFETY (secsgem.secs.dataitems.ALCD attribute), 72
 PPBODY (class in secsgem.secs.dataitems), 108
 PPGNT (class in secsgem.secs.dataitems), 109
 PPID (class in secsgem.secs.dataitems), 109
 PPID_NOT_FOUND (secsgem.secs.dataitems.ACKC7 attribute), 70

- PRAXI (*class in secsgem.secs.dataitems*), 110
- PRDCT (*class in secsgem.secs.dataitems*), 111
- preferredTypes (*secsgem.secs.variables.SecsVarArray attribute*), 46
- preferredTypes (*secsgem.secs.variables.SecsVarBinary attribute*), 47
- preferredTypes (*secsgem.secs.variables.SecsVarBoolean attribute*), 48
- preferredTypes (*secsgem.secs.variables.SecsVarF4 attribute*), 62
- preferredTypes (*secsgem.secs.variables.SecsVarF8 attribute*), 60
- preferredTypes (*secsgem.secs.variables.SecsVarI1 attribute*), 56
- preferredTypes (*secsgem.secs.variables.SecsVarI2 attribute*), 58
- preferredTypes (*secsgem.secs.variables.SecsVarI4 attribute*), 59
- preferredTypes (*secsgem.secs.variables.SecsVarI8 attribute*), 55
- preferredTypes (*secsgem.secs.variables.SecsVarJIS8 attribute*), 52
- preferredTypes (*secsgem.secs.variables.SecsVarList attribute*), 44
- preferredTypes (*secsgem.secs.variables.SecsVarString attribute*), 51
- preferredTypes (*secsgem.secs.variables.SecsVarU1 attribute*), 64
- preferredTypes (*secsgem.secs.variables.SecsVarU2 attribute*), 66
- preferredTypes (*secsgem.secs.variables.SecsVarU4 attribute*), 68
- preferredTypes (*secsgem.secs.variables.SecsVarU8 attribute*), 63
- PRESENT (*secsgem.secs.dataitems.ATTRRELN attribute*), 76
- ## R
- RCMD (*class in secsgem.secs.dataitems*), 111
- REFP (*class in secsgem.secs.dataitems*), 112
- register_stream_function() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 209
- register_stream_function() (*secsgem.gem.handler.GemHandler method*), 196
- register_stream_function() (*secsgem.gem.hosthandler.GemHostHandler method*), 202
- register_stream_function() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 188
- registered_collection_events (*secsgem.gem.equipmenthandler.GemEquipmentHandler attribute*), 207
- registered_reports (*secsgem.gem.equipmenthandler.GemEquipmentHandler attribute*), 207
- RELATION (*secsgem.secs.functions.SecsS14F01 attribute*), 184
- remote_commands (*secsgem.gem.equipmenthandler.GemEquipmentHandler attribute*), 208
- remote_commands (*secsgem.gem.handler.GemHandler attribute*), 196
- remote_commands (*secsgem.gem.hosthandler.GemHostHandler attribute*), 202
- remote_commands (*secsgem.secs.handler.SecsHandler attribute*), 189
- remove_peer() (*secsgem.hsms.connectionmanager.HsmsConnectionManager method*), 40
- reports (*secsgem.gem.equipmenthandler.CollectionEventLink attribute*), 213
- request_ec() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- request_ec() (*secsgem.gem.handler.GemHandler method*), 197
- request_ec() (*secsgem.gem.hosthandler.GemHostHandler method*), 203
- request_ec() (*secsgem.secs.handler.SecsHandler method*), 191
- request_ecs() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- request_ecs() (*secsgem.gem.handler.GemHandler method*), 197
- request_ecs() (*secsgem.gem.hosthandler.GemHostHandler method*), 203
- request_ecs() (*secsgem.secs.handler.SecsHandler method*), 190
- request_process_program() (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- request_process_program() (*secsgem.gem.handler.GemHandler method*),

- 194
- `request_process_program()` (*secsgem.gem.hosthandler.GemHostHandler method*), 203
- `request_sv()` (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- `request_sv()` (*secsgem.gem.handler.GemHandler method*), 197
- `request_sv()` (*secsgem.gem.hosthandler.GemHostHandler method*), 203
- `request_sv()` (*secsgem.secs.handler.SecsHandler method*), 190
- `request_svcs()` (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- `request_svcs()` (*secsgem.gem.handler.GemHandler method*), 197
- `request_svcs()` (*secsgem.gem.hosthandler.GemHostHandler method*), 203
- `request_svcs()` (*secsgem.secs.handler.SecsHandler method*), 190
- ROW (*secsgem.secs.dataitems.MAPFT attribute*), 101
- ROWCT (*class in secsgem.secs.dataitems*), 112
- ROWS_BOT_DECR (*secsgem.secs.dataitems.PRAXI attribute*), 111
- ROWS_BOT_INCR (*secsgem.secs.dataitems.PRAXI attribute*), 111
- ROWS_TOP_DECR (*secsgem.secs.dataitems.PRAXI attribute*), 111
- ROWS_TOP_INCR (*secsgem.secs.dataitems.PRAXI attribute*), 111
- RPSEL (*class in secsgem.secs.dataitems*), 112
- RPTID (*class in secsgem.secs.dataitems*), 113
- RPTID_REDEFINED (*secsgem.secs.dataitems.DRACK attribute*), 84
- RPTID_UNKNOWN (*secsgem.secs.dataitems.LRACK attribute*), 100
- RSINF (*class in secsgem.secs.dataitems*), 114
- ## S
- SDACK (*class in secsgem.secs.dataitems*), 114
- SDBIN (*class in secsgem.secs.dataitems*), 114
- `secs_decode()` (*secsgem.gem.equipmenthandler.GemEquipmentHandler method*), 210
- `secs_decode()` (*secsgem.gem.handler.GemHandler method*), 197
- `secs_decode()` (*secsgem.gem.hosthandler.GemHostHandler method*), 204
- `secs_decode()` (*secsgem.secs.handler.SecsHandler method*), 192
- `secsgem.common` (*module*), 213
- `secsgem.secs.dataitems` (*module*), 68
- `secsgem.secs.functionbase` (*module*), 122
- `secsgem.secs.functions` (*module*), 122
- `secsgem.secs.variables` (*module*), 40
- SecsHandler (*class in secsgem.secs.handler*), 188
- SecsS00F00 (*class in secsgem.secs.functions*), 122
- SecsS01F00 (*class in secsgem.secs.functions*), 123
- SecsS01F01 (*class in secsgem.secs.functions*), 123
- SecsS01F02 (*class in secsgem.secs.functions*), 123
- SecsS01F03 (*class in secsgem.secs.functions*), 124
- SecsS01F04 (*class in secsgem.secs.functions*), 124
- SecsS01F11 (*class in secsgem.secs.functions*), 125
- SecsS01F12 (*class in secsgem.secs.functions*), 126
- SecsS01F13 (*class in secsgem.secs.functions*), 126
- SecsS01F14 (*class in secsgem.secs.functions*), 127
- SecsS01F15 (*class in secsgem.secs.functions*), 128
- SecsS01F16 (*class in secsgem.secs.functions*), 128
- SecsS01F17 (*class in secsgem.secs.functions*), 128
- SecsS01F18 (*class in secsgem.secs.functions*), 129
- SecsS02F00 (*class in secsgem.secs.functions*), 129
- SecsS02F13 (*class in secsgem.secs.functions*), 129
- SecsS02F14 (*class in secsgem.secs.functions*), 130
- SecsS02F15 (*class in secsgem.secs.functions*), 130
- SecsS02F16 (*class in secsgem.secs.functions*), 131
- SecsS02F17 (*class in secsgem.secs.functions*), 132
- SecsS02F18 (*class in secsgem.secs.functions*), 132
- SecsS02F29 (*class in secsgem.secs.functions*), 132
- SecsS02F30 (*class in secsgem.secs.functions*), 133
- SecsS02F33 (*class in secsgem.secs.functions*), 134
- SecsS02F34 (*class in secsgem.secs.functions*), 135
- SecsS02F35 (*class in secsgem.secs.functions*), 135
- SecsS02F36 (*class in secsgem.secs.functions*), 136
- SecsS02F37 (*class in secsgem.secs.functions*), 137
- SecsS02F38 (*class in secsgem.secs.functions*), 137
- SecsS02F41 (*class in secsgem.secs.functions*), 138
- SecsS02F42 (*class in secsgem.secs.functions*), 138
- SecsS05F00 (*class in secsgem.secs.functions*), 139
- SecsS05F01 (*class in secsgem.secs.functions*), 140
- SecsS05F02 (*class in secsgem.secs.functions*), 140
- SecsS05F03 (*class in secsgem.secs.functions*), 141
- SecsS05F04 (*class in secsgem.secs.functions*), 141
- SecsS05F05 (*class in secsgem.secs.functions*), 142
- SecsS05F06 (*class in secsgem.secs.functions*), 142
- SecsS05F07 (*class in secsgem.secs.functions*), 143
- SecsS05F08 (*class in secsgem.secs.functions*), 143
- SecsS05F09 (*class in secsgem.secs.functions*), 144
- SecsS05F10 (*class in secsgem.secs.functions*), 145
- SecsS05F11 (*class in secsgem.secs.functions*), 145
- SecsS05F12 (*class in secsgem.secs.functions*), 146
- SecsS05F13 (*class in secsgem.secs.functions*), 146
- SecsS05F14 (*class in secsgem.secs.functions*), 147

- SecsS05F15 (class in secsgem.secs.functions), 147
- SecsS05F16 (class in secsgem.secs.functions), 148
- SecsS05F17 (class in secsgem.secs.functions), 149
- SecsS05F18 (class in secsgem.secs.functions), 149
- SecsS06F00 (class in secsgem.secs.functions), 150
- SecsS06F05 (class in secsgem.secs.functions), 150
- SecsS06F06 (class in secsgem.secs.functions), 151
- SecsS06F07 (class in secsgem.secs.functions), 151
- SecsS06F08 (class in secsgem.secs.functions), 151
- SecsS06F11 (class in secsgem.secs.functions), 153
- SecsS06F12 (class in secsgem.secs.functions), 154
- SecsS06F15 (class in secsgem.secs.functions), 154
- SecsS06F16 (class in secsgem.secs.functions), 154
- SecsS06F19 (class in secsgem.secs.functions), 155
- SecsS06F20 (class in secsgem.secs.functions), 156
- SecsS06F21 (class in secsgem.secs.functions), 156
- SecsS06F22 (class in secsgem.secs.functions), 157
- SecsS07F00 (class in secsgem.secs.functions), 157
- SecsS07F01 (class in secsgem.secs.functions), 158
- SecsS07F02 (class in secsgem.secs.functions), 158
- SecsS07F03 (class in secsgem.secs.functions), 159
- SecsS07F04 (class in secsgem.secs.functions), 159
- SecsS07F05 (class in secsgem.secs.functions), 160
- SecsS07F06 (class in secsgem.secs.functions), 160
- SecsS07F17 (class in secsgem.secs.functions), 161
- SecsS07F18 (class in secsgem.secs.functions), 161
- SecsS07F19 (class in secsgem.secs.functions), 162
- SecsS07F20 (class in secsgem.secs.functions), 162
- SecsS09F00 (class in secsgem.secs.functions), 162
- SecsS09F01 (class in secsgem.secs.functions), 163
- SecsS09F03 (class in secsgem.secs.functions), 163
- SecsS09F05 (class in secsgem.secs.functions), 163
- SecsS09F07 (class in secsgem.secs.functions), 164
- SecsS09F09 (class in secsgem.secs.functions), 164
- SecsS09F11 (class in secsgem.secs.functions), 165
- SecsS09F13 (class in secsgem.secs.functions), 165
- SecsS10F00 (class in secsgem.secs.functions), 166
- SecsS10F01 (class in secsgem.secs.functions), 166
- SecsS10F02 (class in secsgem.secs.functions), 166
- SecsS10F03 (class in secsgem.secs.functions), 167
- SecsS10F04 (class in secsgem.secs.functions), 167
- SecsS12F00 (class in secsgem.secs.functions), 168
- SecsS12F01 (class in secsgem.secs.functions), 168
- SecsS12F02 (class in secsgem.secs.functions), 170
- SecsS12F03 (class in secsgem.secs.functions), 170
- SecsS12F04 (class in secsgem.secs.functions), 171
- SecsS12F05 (class in secsgem.secs.functions), 173
- SecsS12F06 (class in secsgem.secs.functions), 173
- SecsS12F07 (class in secsgem.secs.functions), 174
- SecsS12F08 (class in secsgem.secs.functions), 175
- SecsS12F09 (class in secsgem.secs.functions), 175
- SecsS12F10 (class in secsgem.secs.functions), 176
- SecsS12F11 (class in secsgem.secs.functions), 176
- SecsS12F12 (class in secsgem.secs.functions), 177
- SecsS12F13 (class in secsgem.secs.functions), 178
- SecsS12F14 (class in secsgem.secs.functions), 178
- SecsS12F15 (class in secsgem.secs.functions), 179
- SecsS12F16 (class in secsgem.secs.functions), 180
- SecsS12F17 (class in secsgem.secs.functions), 180
- SecsS12F18 (class in secsgem.secs.functions), 181
- SecsS12F19 (class in secsgem.secs.functions), 182
- SecsS14F00 (class in secsgem.secs.functions), 182
- SecsS14F01 (class in secsgem.secs.functions), 183
- SecsS14F02 (class in secsgem.secs.functions), 184
- SecsS14F03 (class in secsgem.secs.functions), 186
- SecsS14F04 (class in secsgem.secs.functions), 186
- SecsStreamFunction (class in secsgem.secs.functionbase), 122
- SecsVar (class in secsgem.secs.variables), 40
- SecsVarArray (class in secsgem.secs.variables), 45
- SecsVarBinary (class in secsgem.secs.variables), 47
- SecsVarBoolean (class in secsgem.secs.variables), 48
- SecsVarDynamic (class in secsgem.secs.variables), 41
- SecsVarF4 (class in secsgem.secs.variables), 61
- SecsVarF8 (class in secsgem.secs.variables), 60
- SecsVarI1 (class in secsgem.secs.variables), 56
- SecsVarI2 (class in secsgem.secs.variables), 57
- SecsVarI4 (class in secsgem.secs.variables), 59
- SecsVarI8 (class in secsgem.secs.variables), 55
- SecsVarJIS8 (class in secsgem.secs.variables), 52
- SecsVarList (class in secsgem.secs.variables), 44
- SecsVarNumber (class in secsgem.secs.variables), 53
- SecsVarString (class in secsgem.secs.variables), 51
- SecsVarText (class in secsgem.secs.variables), 49
- SecsVarU1 (class in secsgem.secs.variables), 64
- SecsVarU2 (class in secsgem.secs.variables), 65
- SecsVarU4 (class in secsgem.secs.variables), 67
- SecsVarU8 (class in secsgem.secs.variables), 63
- selectTimeout (secsgem.hsms.connections.HsmsActiveConnection attribute), 35
- selectTimeout (secsgem.hsms.connections.HsmsConnection attribute), 35
- selectTimeout (secsgem.hsms.connections.HsmsMultiPassiveConnection attribute), 37
- selectTimeout (secsgem.hsms.connections.HsmsMultiPassiveServer attribute), 37
- selectTimeout (secsgem.hsms.connections.HsmsPassiveConnection attribute), 36
- SEND (secsgem.secs.dataitems.SDBIN attribute), 115
- send_and_waitfor_response() (secsgem.gem.equipmenthandler.GemEquipmentHandler

<i>method</i>), 210		191	
<code>send_and_waitfor_response()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 197	<code>send_linktest_req()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 211
<code>send_and_waitfor_response()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204	<code>send_linktest_req()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 198
<code>send_and_waitfor_response()</code>	(secs- <i>gem.hsms.handler.HsmsHandler</i> method), 38	<code>send_linktest_req()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204
<code>send_and_waitfor_response()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method), 192	<code>send_linktest_req()</code>	(secs- <i>gem.hsms.handler.HsmsHandler</i> method), 39
<code>send_deselect_req()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 210	<code>send_linktest_req()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method), 193
<code>send_deselect_req()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 197	<code>send_linktest_rsp()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 211
<code>send_deselect_req()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204	<code>send_linktest_rsp()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 198
<code>send_deselect_req()</code>	(secs- <i>gem.hsms.handler.HsmsHandler</i> method), 39	<code>send_linktest_rsp()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204
<code>send_deselect_req()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method), 192	<code>send_linktest_rsp()</code>	(secs- <i>gem.hsms.handler.HsmsHandler</i> method), 39
<code>send_deselect_rsp()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 211	<code>send_linktest_rsp()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method), 193
<code>send_deselect_rsp()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 198	<code>send_packet()</code>	(secs- <i>gem.hsms.connections.HsmsActiveConnection</i> method), 36
<code>send_deselect_rsp()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204	<code>send_packet()</code>	(secs- <i>gem.hsms.connections.HsmsConnection</i> method), 35
<code>send_deselect_rsp()</code>	(secs- <i>gem.hsms.handler.HsmsHandler</i> method), 39	<code>send_packet()</code>	(secs- <i>gem.hsms.connections.HsmsMultiPassiveConnection</i> method), 37
<code>send_deselect_rsp()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method), 192	<code>send_packet()</code>	(secs- <i>gem.hsms.connections.HsmsPassiveConnection</i> method), 36
<code>send_equipment_terminal()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 211	<code>send_process_program()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i> method), 211
<code>send_equipment_terminal()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 198	<code>send_process_program()</code>	(secs- <i>gem.gem.handler.GemHandler</i> method), 194
<code>send_equipment_terminal()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204	<code>send_process_program()</code>	(secs- <i>gem.gem.hosthandler.GemHostHandler</i> method), 204
<code>send_equipment_terminal()</code>	(secs- <i>gem.secs.handler.SecsHandler</i> method),	<code>send_reject_rsp()</code>	(secs- <i>gem.gem.equipmenthandler.GemEquipmentHandler</i>

<i>method</i>), 211		<i>method</i>), 205	
send_reject_rsp()	(secs-	send_select_rsp()	(secs-
gem.gem.handler.GemHandler	method),	gem.hsms.handler.HsmsHandler	method),
198		39	
send_reject_rsp()	(secs-	send_select_rsp()	(secs-
gem.gem.hosthandler.GemHostHandler	method),	gem.secs.handler.SecsHandler	method),
method), 204		193	
send_reject_rsp()	(secs-	send_separate_req()	(secs-
gem.hsms.handler.HsmsHandler	method),	gem.gem.equipmenthandler.GemEquipmentHandler	
39		method), 212	
send_reject_rsp()	(secs-	send_separate_req()	(secs-
gem.secs.handler.SecsHandler	method),	gem.gem.handler.GemHandler	method),
193		198	
send_remote_command()	(secs-	send_separate_req()	(secs-
gem.gem.hosthandler.GemHostHandler		gem.gem.hosthandler.GemHostHandler	
method), 199		method), 205	
send_response()	(secs-	send_separate_req()	(secs-
gem.gem.equipmenthandler.GemEquipmentHandler		gem.hsms.handler.HsmsHandler	method),
method), 211		39	
send_response()	(secs-	send_separate_req()	(secs-
gem.gem.handler.GemHandler	method),	gem.secs.handler.SecsHandler	method),
198		193	
send_response()	(secs-	send_stream_function()	(secs-
gem.gem.hosthandler.GemHostHandler		gem.gem.equipmenthandler.GemEquipmentHandler	
method), 205		method), 212	
send_response()	(secs-	send_stream_function()	(secs-
gem.hsms.handler.HsmsHandler	method),	gem.gem.handler.GemHandler	method),
38		198	
send_response()	(secs-	send_stream_function()	(secs-
gem.secs.handler.SecsHandler	method),	gem.gem.hosthandler.GemHostHandler	
193		method), 205	
send_select_req()	(secs-	send_stream_function()	(secs-
gem.gem.equipmenthandler.GemEquipmentHandler		gem.hsms.handler.HsmsHandler	method),
method), 211		38	
send_select_req()	(secs-	send_stream_function()	(secs-
gem.gem.handler.GemHandler	method),	gem.secs.handler.SecsHandler	method),
198		193	
send_select_req()	(secs-	sendBlockSize	(secs-
gem.gem.hosthandler.GemHostHandler		gem.hsms.connections.HsmsActiveConnection	
method), 205		attribute), 35	
send_select_req()	(secs-	sendBlockSize	(secs-
gem.hsms.handler.HsmsHandler	method),	gem.hsms.connections.HsmsConnection	
39		attribute), 35	
send_select_req()	(secs-	sendBlockSize	(secs-
gem.secs.handler.SecsHandler	method),	gem.hsms.connections.HsmsMultiPassiveConnection	
193		attribute), 37	
send_select_rsp()	(secs-	sendBlockSize	(secs-
gem.gem.equipmenthandler.GemEquipmentHandler		gem.hsms.connections.HsmsPassiveConnection	
method), 211		attribute), 36	
send_select_rsp()	(secs-	set()	(secsgem.secs.functionbase.SecsStreamFunction
gem.gem.handler.GemHandler	method),	method), 122	
198		set()	(secsgem.secs.variables.ANYVALUE method), 44
send_select_rsp()	(secs-	set()	(secsgem.secs.variables.SecsVar method), 41
gem.gem.hosthandler.GemHostHandler		set()	(secsgem.secs.variables.SecsVarArray method),

46
 set () (secsgem.secs.variables.SecsVarBinary method), 47
 set () (secsgem.secs.variables.SecsVarBoolean method), 48
 set () (secsgem.secs.variables.SecsVarDynamic method), 41
 set () (secsgem.secs.variables.SecsVarF4 method), 63
 set () (secsgem.secs.variables.SecsVarF8 method), 61
 set () (secsgem.secs.variables.SecsVarI1 method), 57
 set () (secsgem.secs.variables.SecsVarI2 method), 59
 set () (secsgem.secs.variables.SecsVarI4 method), 60
 set () (secsgem.secs.variables.SecsVarI8 method), 56
 set () (secsgem.secs.variables.SecsVarJIS8 method), 53
 set () (secsgem.secs.variables.SecsVarList method), 45
 set () (secsgem.secs.variables.SecsVarNumber method), 54
 set () (secsgem.secs.variables.SecsVarString method), 52
 set () (secsgem.secs.variables.SecsVarText method), 50
 set () (secsgem.secs.variables.SecsVarU1 method), 65
 set () (secsgem.secs.variables.SecsVarU2 method), 67
 set () (secsgem.secs.variables.SecsVarU4 method), 68
 set () (secsgem.secs.variables.SecsVarU8 method), 64
 set_alarm () (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 208
 set_ec () (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 212
 set_ec () (secsgem.gem.handler.GemHandler method), 199
 set_ec () (secsgem.gem.hosthandler.GemHostHandler method), 205
 set_ec () (secsgem.secs.handler.SecsHandler method), 191
 set_ecs () (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 212
 set_ecs () (secsgem.gem.handler.GemHandler method), 199
 set_ecs () (secsgem.gem.hosthandler.GemHostHandler method), 205
 set_ecs () (secsgem.secs.handler.SecsHandler method), 191
 SHEAD (class in secsgem.secs.dataitems), 115
 SOFTREV (class in secsgem.secs.dataitems), 115
 SOFTREV (secsgem.gem.handler.GemHandler attribute), 194
 start () (secsgem.hsms.connections.HsmsMultiPassiveServer method), 37
 status_variables (secsgem.gem.equipmenthandler.GemEquipmentHandler attribute), 206
 StatusVariable (class in secsgem.gem.equipmenthandler), 212
 stop () (secsgem.hsms.connectionmanager.HsmsConnectionManager method), 40
 stop () (secsgem.hsms.connections.HsmsMultiPassiveServer method), 37
 stream_function () (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 212
 stream_function () (secsgem.gem.handler.GemHandler method), 199
 stream_function () (secsgem.gem.hosthandler.GemHostHandler method), 205
 stream_function () (secsgem.secs.handler.SecsHandler method), 191
 STRP (class in secsgem.secs.dataitems), 115
 StructureDisplayingMeta (class in secsgem.secs.functionbase), 122
 subscribe_collection_event () (secsgem.gem.hosthandler.GemHostHandler method), 199
 SUCCESSFUL (secsgem.secs.dataitems.OBJACK attribute), 105
 supports_value () (secsgem.secs.variables.SecsVarBinary method), 47
 supports_value () (secsgem.secs.variables.SecsVarBoolean method), 48
 supports_value () (secsgem.secs.variables.SecsVarF4 method), 63
 supports_value () (secsgem.secs.variables.SecsVarF8 method), 61
 supports_value () (secsgem.secs.variables.SecsVarI1 method), 57
 supports_value () (secsgem.secs.variables.SecsVarI2 method), 59
 supports_value () (secsgem.secs.variables.SecsVarI4 method), 60
 supports_value () (secsgem.secs.variables.SecsVarI8 method), 56
 supports_value () (secsgem.secs.variables.SecsVarJIS8 method), 53
 supports_value () (secsgem.secs.variables.SecsVarNumber method), 54
 supports_value () (secsgem.secs.variables.SecsVarString method), 52
 supports_value () (secsgem.secs.variables.SecsVarText method), 50

50
 supports_value() (secsgem.secs.variables.SecsVarU1 method), 65
 supports_value() (secsgem.secs.variables.SecsVarU2 method), 67
 supports_value() (secsgem.secs.variables.SecsVarU4 method), 68
 supports_value() (secsgem.secs.variables.SecsVarU8 method), 64
 SV (class in secsgem.secs.dataitems), 116
 SVID (class in secsgem.secs.dataitems), 116
 SVNAME (class in secsgem.secs.dataitems), 117

T

T3 (secsgem.hsms.connections.HsmsActiveConnection attribute), 35
 T3 (secsgem.hsms.connections.HsmsConnection attribute), 35
 T3 (secsgem.hsms.connections.HsmsMultiPassiveConnection attribute), 37
 T3 (secsgem.hsms.connections.HsmsPassiveConnection attribute), 36
 T5 (secsgem.hsms.connections.HsmsActiveConnection attribute), 35
 T5 (secsgem.hsms.connections.HsmsConnection attribute), 35
 T5 (secsgem.hsms.connections.HsmsMultiPassiveConnection attribute), 37
 T5 (secsgem.hsms.connections.HsmsPassiveConnection attribute), 36
 T6 (secsgem.hsms.connections.HsmsActiveConnection attribute), 35
 T6 (secsgem.hsms.connections.HsmsConnection attribute), 35
 T6 (secsgem.hsms.connections.HsmsMultiPassiveConnection attribute), 37
 T6 (secsgem.hsms.connections.HsmsPassiveConnection attribute), 36
 TERMINAL_NOT_AVAILABLE (secsgem.secs.dataitems.ACKC10 attribute), 71
 TEXT (class in secsgem.secs.dataitems), 117
 textCode (secsgem.secs.variables.SecsVarArray attribute), 46
 textCode (secsgem.secs.variables.SecsVarBinary attribute), 47
 textCode (secsgem.secs.variables.SecsVarBoolean attribute), 48
 textCode (secsgem.secs.variables.SecsVarF4 attribute), 62
 textCode (secsgem.secs.variables.SecsVarF8 attribute), 60
 textCode (secsgem.secs.variables.SecsVarI1 attribute), 56
 textCode (secsgem.secs.variables.SecsVarI2 attribute), 58
 textCode (secsgem.secs.variables.SecsVarI4 attribute), 59
 textCode (secsgem.secs.variables.SecsVarI8 attribute), 55
 textCode (secsgem.secs.variables.SecsVarJIS8 attribute), 52
 textCode (secsgem.secs.variables.SecsVarList attribute), 44
 textCode (secsgem.secs.variables.SecsVarNumber attribute), 54
 textCode (secsgem.secs.variables.SecsVarString attribute), 51
 textCode (secsgem.secs.variables.SecsVarText attribute), 49
 textCode (secsgem.secs.variables.SecsVarU1 attribute), 64
 textCode (secsgem.secs.variables.SecsVarU2 attribute), 66
 textCode (secsgem.secs.variables.SecsVarU4 attribute), 67
 textCode (secsgem.secs.variables.SecsVarU8 attribute), 63
 TID (class in secsgem.secs.dataitems), 118
 TIME (class in secsgem.secs.dataitems), 118
 TIMESTAMP (class in secsgem.secs.dataitems), 118
 trigger_collection_events() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 207

U

UNITS (class in secsgem.secs.dataitems), 119
 UNKNOWN_ID (secsgem.secs.dataitems.MDACK attribute), 101
 UNKNOWN_MAP_FORMAT (secsgem.secs.dataitems.GRNT1 attribute), 97
 unregister_stream_function() (secsgem.gem.equipmenthandler.GemEquipmentHandler method), 212
 unregister_stream_function() (secsgem.gem.handler.GemHandler method), 199
 unregister_stream_function() (secsgem.gem.hosthandler.GemHostHandler method), 205
 unregister_stream_function() (secsgem.secs.handler.SecsHandler method), 188

UPPER_LEFT (*secsgem.secs.dataitems.ORLOC* attribute), 108
UPPER_RIGHT (*secsgem.secs.dataitems.ORLOC* attribute), 108

V

V (*class in secsgem.secs.dataitems*), 119
VID (*class in secsgem.secs.dataitems*), 120
VID_UNKNOWN (*secsgem.secs.dataitems.DRACK* attribute), 84

W

WAFER (*secsgem.secs.dataitems.IDTYP* attribute), 98
WAFER_CASSETTE (*secsgem.secs.dataitems.IDTYP* attribute), 99
waitfor_communicating() (*secsgem.gem.equipmenthandler.GemEquipmentHandler* method), 212
waitfor_communicating() (*secsgem.gem.handler.GemHandler* method), 194
waitfor_communicating() (*secsgem.gem.hosthandler.GemHostHandler* method), 206
WILL_NOT_ACCEPT (*secsgem.secs.dataitems.PPGNT* attribute), 109

X

XDIES (*class in secsgem.secs.dataitems*), 120
XYPOS (*class in secsgem.secs.dataitems*), 121

Y

YDIES (*class in secsgem.secs.dataitems*), 121