
Pytanga

Renato Almeida de Oliveira

Jan 03, 2021

USING PYTANGA

1 Quick Start	3
1.1 Installing	3
1.2 Basic Usage	3
1.3 Configuring BGP	4
1.4 Configuring Prefix-List	6
1.5 Configuring IP Interface with OpenConfig	7
2 Component Creation	9
2.1 Implementation Example	9
2.2 Using the new Module	11
3 Components	13
3.1 Cisco	13
3.2 OpenConfig	28
3.3 AbstractComponent module	37
3.4 Config module	37
4 Helpers	39
4.1 Cisco IOS-XE	39
4.2 OpenConfig	40
5 Visitors	41
5.1 AbstractVisitor module	41
5.2 NETCONFvisitor module	41
6 Indices and tables	43
Python Module Index	45
Index	47

Pytanga is a Python library that aims to simplify YANG payload creation, its architecture is based on the Composite and Visitor design patterns.

Similar to YANG models where a container can have leaves and other containers, building a part-whole architecture. Pitanga modules define a component with attributes (leaves), and children that represent the inner containers of the module.

For the data output, Pytanga implements a Visitor Pattern that is injected in each component and build the desired output, currently implemented only for NETCONF.

With that architecture, it is possible to define the YANG models' logic and syntax tests decoupled of the payload generation.

QUICK START

1.1 Installing

```
pip install pytanga
```

1.2 Basic Usage

Pytanga uses a Composite pattern to abstract YANG models, so each component has an add method which is used to compose the payload.

Therefore, to build the desired payload it will be necessary to instantiate all modules equivalent to the YANG models and use the add method building the necessary hierarchy, as the example below:

```
from pytanga.components import configComponent
from pytanga.components.OpenConfig.routing import networkInstancesComponent
from pytanga.components.OpenConfig.routing import networkInstanceComponent
from pytanga.components.OpenConfig.routing import protocolsComponent
from pytanga.components.OpenConfig.routing import protocolComponent
from pytanga.components.OpenConfig.routing.static import staticroutesComponent
from pytanga.components.OpenConfig.routing.static import staticComponent
from pytanga.components.OpenConfig.routing.static import nexthopsComponent
from pytanga.components.OpenConfig.routing.static import nexthopComponent
from pytanga.components.OpenConfig.routing.static import interfacerefComponent
from pytanga.visitors import NETCONFVisitor
from xml.dom.minidom import parseString

config = configComponent()
netInsts = networkInstancesComponent()
netinst = networkInstanceComponent()
protos = protocolsComponent()
proto = protocolComponent(identifier='STATIC', name='DEFAULT')
staticroutes = staticroutesComponent()
static = staticComponent(prefix='172.30.0.0/24')
nexthops = nexthopsComponent()
nexthop = nexthopComponent(index='NETCONF', next_hop='192.168.0.4')

nexthops.add(nexthop)
static.add(nexthops)
staticroutes.add(static)
proto.add(staticroutes)
protos.add(proto)
```

(continues on next page)

(continued from previous page)

```

netinst.add(protos)
netInsts.add(netinst)
config.add(netInsts)

serializer = NETCONFVisitor()
output = static.parse(serializer)
xml_string = serializer.print(output)
print(parseString(xml_string).toprettyxml())

```

Resulting in the following output

```

<static>
  <prefix>172.30.0.0/24</prefix>
  <config>
    <prefix>172.30.0.0/24</prefix>
  </config>
  <next-hops>
    <next-hop>
      <index>NETCONF</index>
      <config>
        <index>NETCONF</index>
        <next-hop>192.168.0.4</next-hop>
      </config>
    </next-hop>
  </next-hops>
</static>

```

1.3 Configuring BGP

For configure BGP use the *ConfigureBGP* Helper

```

from pytanga.components import configComponent
from pytanga.components.Cisco.xe import nativeComponent
from pytanga.components.Cisco.xe import routerComponent
from pytanga.helpers.Cisco.xe import ConfigureBGP
from pytanga.visitors import NETCONFVisitor
from xml.dom.minidom import parseString

BGPHelper = ConfigureBGP(asn=100 , router_id='10.0.0.2')
BGPHelper.addAfi_Safi(afi_name = 'ipv4' , safi_name='unicast')
BGPHelper.addNeighbor(address= '10.0.0.1' , remote_as='100')
BGPHelper.addNeighbor(address= '10.0.0.3' , remote_as='100')
BGPHelper.addAfiSafiNeighbor(afi_safi='ipv4-unicast' , address= '10.0.0.1')
BGPHelper.addAfiSafiNeighbor(afi_safi='ipv4-unicast' , address= '10.0.0.3')
BGPHelper.addAfiSafiNeighborRouteMap(afi_safi='ipv4-unicast' , nei_address= '10.0.0.3
↪' , inout='in' , name='Teste')
BGPHelper.addAfiSafiNetwork(afi_safi='ipv4-unicast' , network="10.0.0.0" , mask='255.
↪255.255.255')
BGP = BGPHelper.getBGP()

router = routerComponent()
XENative = nativeComponent()
config = configComponent()

```

(continues on next page)

(continued from previous page)

```

router.add(BGP)
XENative.add(router)
config.add(XENative)

serializer = NETCONFVisitor()
output = config.parse(serializer)
xml_string = serializer.print(output)
print(parseString(xml_string).toprettyxml())

```

Resulting in the following output

```

<config>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <router>
      <bgp xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-bgp">
        <id>100</id>
        <bgp>
          <router-id>
            <ip-id>10.0.0.1</ip-id>
          </router-id>
        </bgp>
        <address-family>
          <no-vrf>
            <ipv4>
              <af-name>unicast</af-name>
              <ipv4-unicast>
                <neighbor>
                  <id>10.0.0.2</id>
                </neighbor>
                <neighbor>
                  <id>10.0.0.3</id>
                  <route-map>
                    <inout>in</inout>
                    <route-map-name>Teste</route-map-name>
                  </route-map>
                </neighbor>
                <network>
                  <with-mask>
                    <number>10.0.0.0</number>
                    <mask>255.255.255.255</mask>
                  </with-mask>
                </network>
              </ipv4-unicast>
            </ipv4>
          </no-vrf>
        </address-family>
        <neighbor>
          <id>10.0.0.2</id>
          <remote-as>100</remote-as>
        </neighbor>
        <neighbor>
          <id>10.0.0.3</id>
          <remote-as>100</remote-as>
        </neighbor>
      </bgp>
    </router>
  </native>

```

(continues on next page)

```
</config>
```

IOS-XE 16.9.1 resulting configuration:

```
router bgp 100
  bgp router-id 10.0.0.2
  bgp log-neighbor-changes
  neighbor 10.0.0.1 remote-as 100
  neighbor 10.0.0.3 remote-as 100
  !
  address-family ipv4
    network 10.0.0.0 mask 255.255.255.255
    neighbor 10.0.0.1 activate
    neighbor 10.0.0.3 activate
    neighbor 10.0.0.3 route-map Teste in
  exit-address-family
```

1.4 Configuring Prefix-List

For configure prefix list use the *ConfigurePrefixList* Helper

```
from pytanga.components import configComponent
from pytanga.components.Cisco.xe import nativeComponent
from pytanga.components.Cisco.xe.ip import ipComponent
from pytanga.helpers.Cisco.xe import ConfigurePrefixList
from pytanga.visitors import NETCONFVisitor
from xml.dom.minidom import parseString

config = configComponent()
native = nativeComponent()
ip = ipComponent()
PrefixListHelper = ConfigurePrefixList(name="TEST-AS")
PrefixListHelper.addPrefix(action="permit" , network="10.0.40.0/24")
PrefixListHelper.addPrefix(action="permit" , network="10.0.50.0/24")

ip.add(PrefixListHelper.getPrefixList())
native.add(ip)
config.add(native)

serializer = NETCONFVisitor()
output = config.parse(serializer)
xml_string = serializer.print(output)
print(parseString(xml_string).toprettyxml())
```

Resulting in the following output

```
<config>
  <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    <ip>
      <prefix-list>
        <prefixes>
          <name>TEST-AS</name>
          <seq>
            <no>5</no>
```

(continues on next page)

(continued from previous page)

```

        <permit>
          <ip>10.0.40.0/24</ip>
        </permit>
      </seq>
    <seq>
      <no>10</no>
      <permit>
        <ip>10.0.50.0/24</ip>
      </permit>
    </seq>
  </prefixes>
</prefix-list>
</ip>
</native>
</config>

```

IOS-XE 16.9.1 resulting configuration:

```

ip prefix-list TEST-AS seq 5 permit 10.0.40.0/24
ip prefix-list TEST-AS seq 10 permit 10.0.50.0/24

```

1.5 Configuring IP Interface with OpenConfig

For configure an IP Interface use the *CreateIPInterface* Helper

```

from pytanga.components import configComponent
from pytanga.helpers.OpenConfig.interfaces import createIPInterface
from pytanga.components.OpenConfig.interfaces import interfacesComponent
from pytanga.visitors import NETCONFVisitor
from xml.dom.minidom import parseString

interfaces = interfacesComponent()
interface = createIPInterface(name="GigabitEthernet2",
                             if_type='ethernet',
                             ip_version=4,
                             address='10.0.0.5',
                             prefix_length=30,
                             if_mtu= 1650,
                             if_description='Test Configuration',
                             enabled=True)

interfaces.add(interface)
config = configComponent()
config.add(interfaces)
serializer = NETCONFVisitor()
output = config.parse(serializer)
xml_string = serializer.print(output)
print(parseString(xml_string).toprettyxml())

```

Resulting in the following output

```

<config>
  <interfaces xmlns="http://openconfig.net/yang/interfaces" xmlns:oc-ip="http://
  ↪openconfig.net/yang/interfaces/ip">

```

(continues on next page)

```
<interface>
  <name>GigabitEthernet2</name>
  <config>
    <description>Test Configuration</description>
    <mtu>1650</mtu>
    <enabled>true</enabled>
    <type xmlns:ianaift="urn:ietf:params:xml:ns:yang:iana-if-type">
↪ ianaift:ethernetCsmacd</type>
  </config>
  <subinterfaces>
    <subinterface>
      <index>0</index>
      <oc-ip:ipv4>
        <oc-ip:addresses>
          <oc-ip:address>
            <oc-ip:ip>10.0.0.5</oc-ip:ip>
            <oc-ip:config>
              <oc-ip:ip>10.0.0.5</oc-ip:ip>
              <oc-ip:prefix-length>30</oc-ip:prefix-length>
            </oc-ip:config>
          </oc-ip:address>
        </oc-ip:addresses>
      </oc-ip:ipv4>
    </subinterface>
  </subinterfaces>
</interface>
</interfaces>
</config>
```

IOS-XE 16.9.1 resulting configuration:

```
interface GigabitEthernet2
  description Test Configuration
  mtu 1650
  ip address 10.0.0.5 255.255.255.252
end
```

COMPONENT CREATION

For create a custom component it is required to create a new implementation of the *AbstractComponent*

In the `__init__` method the class should receive all the desired attributes for the YANG module, and pass then to the `setAttributes` method that will perform the data validation and construction of the modules attribute schema.

Set the attribute `self.tag` to the model desired tag, and if required set `self.xmlns` to the XMLNS with the following syntax:

```
self.xmlns = {  
    'xmlns' : "http://cisco.com/ns/yang/Cisco-IOS-XE-native" ,  
}
```

Warning: The NETCONFVisitor only supports three levels of attributes nesting. If you need more than four levels consider building a new module.

2.1 Implementation Example

```
from pytanga.components import AbstractComponent  
  
class testComponent(AbstractComponent):  
  
    def __init__(self, attribute1 , attribute2):  
        self.xmlns = {}  
        self.attributes = self.setAttributes(attribute1 , attribute2)  
        self.parent_xmlns = {}  
        self._children: List[AbstractComponent] = []  
        self.childrenData = []  
        self.tag = 'test'  
  
    @property  
    def xmlns(self):  
        return self.xmlns  
  
    @xmlns.setter  
    def xmlns(self, xmlns):  
        self.xmlns = xmlns  
  
    def setAttributes(self, attribute1 , attribute2):  
        attributes = {}
```

(continues on next page)

```
attributes['l1_var1'] = attribute1
attributes['l1_var2'] = 'var2'
attributes['l1_var3'] = None
attributes['l1_var4'] = {
    'keys': {
        'attr_key' : attribute2
    },
    'value' : 'var4'
}
attributes['level2'] = {
    'l2_var1': 'l2_var1',
    'l2_var2': 'l2_var2',
    'l2_var3': None,
    'l2_var4': {
        'keys': {
            'testkey': "value"
        },
        'value' : 'var4'
    }
}
attributes['level3'] = {
    'level3': {
        'l3_var1': 'l3_var1',
        'l3_var2' : None,
        'l3_var3' : {
            'keys' : {
                'l3_key' : 'key',
            },
            'value' : 'var3'
        }
    }
}

return attributes

def add(self, component) -> None:
    self._children.append(component)

def remove(self, component) -> None:
    self._children.remove(component)

def is_composite(self) -> bool:
    return False

def getXMLNS(self):
    childrenData = []
    for child in self._children:
        self.parent_xmlns.update(child.getXMLNS())
    return self.parent_xmlns

def parse(self, serializer):
    self.childrenData = []
    self.getXMLNS()
    for child in self._children:
        self.childrenData.append(child.parse(serializer))
    return serializer.parse(self)
```

2.2 Using the new Module

```
from newmodule import testComponent
from pytanga.visitors import NETCONFVisitor
from xml.dom.minidom import parseString

module = testComponent("Value1" , "Value2")
serializer = NETCONFVisitor()
output = module.parse(serializer)
xml_string = serializer.print(output)
print(parseString(xml_string).toprettyxml())
```

Resulting in the following output

```
<test>
  <l1_var1>Value1</l1_var1>
  <l1_var2>var2</l1_var2>
  <l1_var3/>
  <l1_var4 attr_key="Value2">var4</l1_var4>
  <level2>
    <l2_var1>l2_var1</l2_var1>
    <l2_var2>l2_var2</l2_var2>
    <l2_var3/>
    <l2_var4 testkey="value">var4</l2_var4>
  </level2>
  <level3>
    <level3>
      <l3_var1>l3_var1</l3_var1>
      <l3_var2/>
      <l3_var3 l3_key="key">var3</l3_var3>
    </level3>
  </level3>
</test>
```


COMPONENTS

3.1 Cisco

3.1.1 XE

native

```
class pytanga.components.Cisco.xe.native.nativeComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent
    add (component) → None
        This should method add a subComponent.
    getXMLNS ()
    is_composite () → bool
    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.
    remove (component) → None
        This should method remove a subComponent.
    property xmlns
```

router

```
class pytanga.components.Cisco.xe.router.routerComponent (operation=None)
    Bases: pytanga.components.AbstractComponent.AbstractComponent
    add (component) → None
        This should method add a subComponent.
    getXMLNS ()
    is_composite () → bool
    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.
    remove (component) → None
        This should method remove a subComponent.
    setAttributes ()
    property xmlns
```

IP

ip

```
class pytanga.components.Cisco.xe.ip.ip.ipComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent
    add (component) → None
        This should method add a subComponent.
    getXMLNS ()
    is_composite () → bool
    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.
    remove (component) → None
        This should method remove a subComponent.
    setAttributes ()
    property xmlns
```

prefix

```
class pytanga.components.Cisco.xe.ip.prefix.prefixComponent (seq, action, network,
                                                         ge=None, le=None)
    Bases: pytanga.components.AbstractComponent.AbstractComponent
    add (component) → None
        This should method add a subComponent.
    getXMLNS ()
    is_composite () → bool
    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.
    remove (component) → None
        This should method remove a subComponent.
    setAttributes (seq, action, network, ge, le)
    property xmlns
exception pytanga.components.Cisco.xe.ip.prefix.prefixSyntaxError
    Bases: Exception
```

prefixlist

```

class pytanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent (name,
                                                                    descrip-
                                                                    tion=None,
                                                                    opera-
                                                                    tion=None)

Bases: pytanga.components.AbstractComponent.AbstractComponent

add (component) → None
    This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (serializer)
    This method should call the parse method for all childrens passing the serializer.

remove (component) → None
    This should method remove a subComponent.

setAttributes (name, description)

property xmlns

```

prefixlists

```

class pytanga.components.Cisco.xe.ip.prefixlists.prefixeslistsComponent (sequence_number=None,
                                                                    op-
                                                                    er-
                                                                    a-
                                                                    tion=None)

Bases: pytanga.components.AbstractComponent.AbstractComponent

add (component) → None
    This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (serializer)
    This method should call the parse method for all childrens passing the serializer.

remove (component) → None
    This should method remove a subComponent.

setAttributes (sequence_number)

property xmlns

```

routemap

```
class pytanga.components.Cisco.xe.ip.routemap.routemapComponent (name)
    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    setAttributes (name)

    property xmlns
```

routemapentry

```
exception pytanga.components.Cisco.xe.ip.routemapentry.routemapSyntaxError
    Bases: Exception

class pytanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent (operation,
                                                                 de-
                                                                 scrip-
                                                                 tion=None,
                                                                 seq_no=None,
                                                                 or-
                                                                 der-
                                                                 ing_seq=None)

    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    setAttributes (operation, description, seq_no, ordering_seq)

    property xmlns
```

BGP

addressFamilyIPv4Unicast

```
class pytanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast.addressFamilyIPv4UnicastComponent
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*auto_summary*, *originate_default*, *default_metric*, *synchronization*, *segment_routing_mpls*)

property xmlns

addressFamilyType

```
class pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent (afi_name,  
safi_name)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*safi_name*)

property xmlns

```
exception pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeSyntaxError
```

Bases: Exception

addressFamilyVRF

```
class pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponent (with_vrf)  
    Bases: pytanga.components.AbstractComponent.AbstractComponent  
  
    add (component) → None  
        This should method add a subComponent.  
  
    getXMLNS ()  
  
    is_composite () → bool  
  
    parse (serializer)  
        This method should call the parse method for all childrens passing the serializer.  
  
    remove (component) → None  
        This should method remove a subComponent.  
  
    setAttributes ()  
  
    property xmlns
```

addressfamilies

```
class pytanga.components.Cisco.xe.bgp.addressfamilies.addressFamiliesComponent  
    Bases: pytanga.components.AbstractComponent.AbstractComponent  
  
    add (component) → None  
        This should method add a subComponent.  
  
    getXMLNS ()  
  
    is_composite () → bool  
  
    parse (serializer)  
        This method should call the parse method for all childrens passing the serializer.  
  
    remove (component) → None  
        This should method remove a subComponent.  
  
    setAttributes ()  
  
    property xmlns
```

bgp

```
class pytanga.components.Cisco.xe.bgp.bgp.bgpComponent (asn,
    aigp_rib_metric=None, always_compare_med=None,
    cluster_id=None, deterministic_med=None,
    enforce_first_as=None, enhanced_error=None,
    fast_external_fallover=None, log_neighbor_changes=None,
    maxas_limit=None, maxcommunity_limit=None,
    route_map_cache=None, update_delay=None,
    router_id=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*asn*)

property **xmlns**

```
class pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent (aigp_rib_metric=None, always_compare_med=None, cluster_id=None, deterministic_med=None, enforce_first_as=None, enhanced_error=None, fast_external_fallover=None, log_neighbor_changes=None, maxas_limit=None, maxcommunity_limit=None, route_map_cache=None, update_delay=None, router_id=None, advertise_best_external=None, dmzlink_bw=None, suppress_inactive=None, soft_reconfig_backup=None, scan_time=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*aigp_rib_metric, always_compare_med, cluster_id, deterministic_med, enforce_first_as, enhanced_error, fast_external_fallover, log_neighbor_changes, maxas_limit, maxcommunity_limit, route_map_cache, update_delay, router_id, advertise_best_external, dmzlink_bw, suppress_inactive, soft_reconfig_backup, scan_time*)

property xmlns

neighbor

```
class pytanga.components.Cisco.xe.bgp.neighbor.neighborComponent (address=None,
                                                                    name=None,
                                                                    re-
                                                                    mote_as=None,
                                                                    clus-
                                                                    ter_id=None,
                                                                    descrip-
                                                                    tion=None,
                                                                    dis-
                                                                    able_connected_check=None,
                                                                    ebgp_multihop=None,
                                                                    pass-
                                                                    word=None,
                                                                    peer_group=None,
                                                                    shut-
                                                                    down=None,
                                                                    keepalive_interval=None,
                                                                    hold-
                                                                    time=None,
                                                                    mini-
                                                                    mum_neighbor_hold=None,
                                                                    ttl_security=None,
                                                                    up-
                                                                    date_source=None,
                                                                    ver-
                                                                    sion=None,
                                                                    acti-
                                                                    vate=None,
                                                                    advertise-
                                                                    ment_interval=None,
                                                                    al-
                                                                    low_policy=None,
                                                                    al-
                                                                    lowas_in=None,
                                                                    de-
                                                                    fault_originate=None,
                                                                    de-
                                                                    fault_originate_route_map=None,
                                                                    dm-
                                                                    zlink_bw=None,
                                                                    maxi-
                                                                    mum_prefix_n=None,
                                                                    maxi-
                                                                    mum_prefix_threshold=None,
                                                                    maxi-
                                                                    mum_prefix_restart=None,
                                                                    maxi-
                                                                    mum_prefix_warning=None,
                                                                    next_hop_self=None,
                                                                    next_hop_self_all=None,
                                                                    next_hop_unchanged=None,
                                                                    route_reflector_client=None,
                                                                    send_community=None,
                                                                    send_label=None,
                                                                    soft_reconfiguration=None,
                                                                    weight=None)
```

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*address, name, remote_as, cluster_id, description, disable_connected_check, ebgp_multihop, password, peer_group, shutdown, keepalive_interval, hold-time, minimum_neighbor_hold, ttl_security, update_source, version, activate, advertisement_interval, allow_policy, allowas_in, default_originate, default_originate_route_map, dmzlink_bw, maximum_prefix_n, maximum_prefix_threshold, maximum_prefix_restart, maximum_prefix_warning, next_hop_self, next_hop_self_all, next_hop_unchanged, route_reflector_client, send_community, send_label, soft_reconfiguration, weight*)

property xmlns

exception `pytanga.components.Cisco.xe.bgp.neighbor.neighborSyntaxError`
Bases: Exception

neighborAdvertiseMap

class `pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent` (*na*

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*name, exist_map, non_exist_map*)

property xmlns

neighborAdvertiseMaps

```
class pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    setAttributes ()

    property xmlns
```

neighborDistributeList

```
class pytanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent

    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    setAttributes (inout, name)

    property xmlns
```

neighborPrefixList

```
class pytanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent (inout, name)

    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.
```

remove (*component*) → None
 This should method remove a subComponent.

setAttributes (*inout, name*)

property xmlns

neighborRouteMap

class `pytanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponent` (*inout, name*)

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
 This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
 This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
 This should method remove a subComponent.

setAttributes (*inout, name*)

property xmlns

network

class `pytanga.components.Cisco.xe.bgp.network.networkComponent` (*network, mask=None, route_map=None, backdoor=None*)

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
 This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
 This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
 This should method remove a subComponent.

setAttributes (*network, mask, route_map, backdoor*)

property xmlns

peergroup

```
class pytanga.components.Cisco.xe.bgp.peergroup.peerGroupComponent (name, remote_as=None, cluster_id=None, description=None, disable_connected_check=None, ebgp_multihop=None, password=None, shutdown=None, keepalive_interval=None, hold_time=None, minimum_neighbor_hold=None, ttl_security=None, update_source=None, version=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes ()

property xmlns

OSPF

Area

```
class pytanga.components.Cisco.xe.ospf.area.areaComponent (area_id, default_fault_cost=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*area_id, default_cost*)

property xmlns

Network

```
class pytanga.components.Cisco.xe.ospf.network.networkComponent (network, wildcard_mask, area=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*ip, wildcard_mask, area*)

property xmlns

ospf

```
class pytanga.components.Cisco.xe.ospf.ospf.ospfComponent (process_id, vrf=None, router_id=None, nsr=None, maximum_paths=None, domain_tag=None, ispf=None, prefix_suppression=None, priority=None, shutdown=None, cost=None, flood_reduction=None, hello_interval=None, mtu_ignore=None, resync_timeout=None, retransmit_interval=None, transmit_delay=None)
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

```
getXMLNS ()  
is_composite () → bool  
parse (serializer)  
    This method should call the parse method for all childrens passing the serializer.  
remove (component) → None  
    This should method remove a subComponent.  
setAttributes (process_id, vrf, router_id, nsr, maximum_paths, domain_tag, ispf, pre-  
    fix_suppression, priority, shutdown, cost, flood_reduction, hello_interval,  
    mtu_ignore, resync_timeout, retransmit_interval, transmit_delay)  
property xmlns
```

3.1.2 XR

Module contents

3.2 OpenConfig

3.2.1 Interfaces

Ethernet

```
class pytanga.components.OpenConfig.interfaces.ethernet.ethernetComponent (mac_address=None,  
    auto_negotiate=None,  
    du-  
    plex_mode=None,  
    port_speed=None,  
    en-  
    able_flow_control=None)  
  
Bases: pytanga.components.AbstractComponent.AbstractComponent  
add (component) → None  
    This should method add a subComponent.  
getXMLNS ()  
is_composite () → bool  
parse (serializer)  
    This method should call the parse method for all childrens passing the serializer.  
remove (component) → None  
    This should method remove a subComponent.  
setAttributes (mac_address, auto_negotiate, duplex_mode, port_speed, enable_flow_control)  
property xmlns
```


Interface

```

class pytanga.components.OpenConfig.interfaces.interface.interfaceComponent (name,
                                                                    if_type,
                                                                    if_description=None,
                                                                    if_mtu=None,
                                                                    en-
                                                                    abled=None)

Bases: pytanga.components.AbstractComponent.AbstractComponent

IETF_INTERFACE_TYPES = {'ethernet': 'ianaift:ethernetCsmacd', 'loopback': 'ianaift:s

add (component) → None
    This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (serializer)
    This method should call the parse method for all childrens passing the serializer.

remove (component) → None
    This should method remove a subComponent.

setAttributes (name, if_type, if_description, if_mtu, enabled)

property xmlns

```

Interfaces

```

class pytanga.components.OpenConfig.interfaces.interfaces.interfacesComponent
Bases: pytanga.components.AbstractComponent.AbstractComponent

add (component) → None
    This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (serializer)
    This method should call the parse method for all childrens passing the serializer.

remove (component) → None
    This should method remove a subComponent.

property xmlns

```

oc_ip

```

class pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponent (version)
Bases: pytanga.components.AbstractComponent.AbstractComponent

add (component) → None
    This should method add a subComponent.

getXMLNS ()

is_composite () → bool

property parent

```

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

property xmlns

oc_ipAddress

class pytanga.components.OpenConfig.interfaces.oc_ipAddress.**oc_ipAddressComponent** (*address,*
pre-
fix_length)

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

property parent

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*address, prefix_length*)

property xmlns

oc_ipAddresses

class pytanga.components.OpenConfig.interfaces.oc_ipAddresses.**oc_ipAddressesComponent**
Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

property parent

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

property xmlns

subinterface

class `pytanga.components.OpenConfig.interfaces.subinterface.subinterfaceComponent` (*index, description=None*)

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*index, description*)

property xmlns

subinterfaces

class `pytanga.components.OpenConfig.interfaces.subinterfaces.subinterfacesComponent`

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

property xmlns

switchedVlan

class `pytanga.components.OpenConfig.interfaces.switchedVlan.switchedVlanComponent` (*interface_name, native_vlan=None, access_vlan=None, trunk_vlans=*

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*interface_mode, native_vlan, access_vlan, trunk_vlans*)

property xmlns

3.2.2 OSPFv2

ospfv2

class `pytanga.components.OpenConfig.routing.ospfv2.ospfv2.ospfv2Component`

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

property xmlns

ospfv2Area

class `pytanga.components.OpenConfig.routing.ospfv2.ospfv2Area.ospfv2AreaComponent` (*identifier*)

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None

This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)

This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None

This should method remove a subComponent.

setAttributes (*identifier*)

property xmlns

ospfv2Areas

```

class pytanga.components.OpenConfig.routing.ospfv2.ospfv2Areas.ospfv2AreasComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    property xmlns

```

ospfv2Global

```

class pytanga.components.OpenConfig.routing.ospfv2.ospfv2Global.ospfv2GlobalComponent (router_
    sum-
    mary_
    igp_sh
    log_aa
    hide_t

    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    setAttributes (router_id, summary_route_cost_mode, igp_shortcuts, log_adjacency_changes,
        hide_transit_only_networks)

    property xmlns

```

ospfv2Interface

```
class pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*if_id, network_type, priority, multi_area_adjacency_primary, authentication_type, metric, passive, hide_network*)

property xmlns

ospfv2Interfaces

```
class pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interfaces.ospfv2InterfacesComponent
```

Bases: *pytanga.components.AbstractComponent.AbstractComponent*

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

property xmlns

3.2.3 Static

interfaceref

class `pytanga.components.OpenConfig.routing.static.interfaceref.interfacerefComponent` (*interfa-
subin-
ter-
face=l*

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*interface, subinterface*)

property xmlns

nexthop

class `pytanga.components.OpenConfig.routing.static.nexthop.nexthopComponent` (*index,
next_hop=None,
met-
ric=None,
re-
curse=None*)

Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*index, next_hop, metric, recurse*)

property xmlns

nexthops

class `pytanga.components.OpenConfig.routing.static.nexthops.nexthopsComponent`
Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

property xmlns

static

class `pytanga.components.OpenConfig.routing.static.static.staticComponent` (*prefix*,
set_tag=None,
de-
scrip-
tion=None,
op-
er-
a-
tion=None)
Bases: `pytanga.components.AbstractComponent.AbstractComponent`

add (*component*) → None
This should method add a subComponent.

getXMLNS ()

is_composite () → bool

parse (*serializer*)
This method should call the parse method for all childrens passing the serializer.

remove (*component*) → None
This should method remove a subComponent.

setAttributes (*prefix*, *set_tag*, *description*)

property xmlns

staticroutes

```

class pytanga.components.OpenConfig.routing.static.staticroutes.staticroutesComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

    remove (component) → None
        This should method remove a subComponent.

    property xmlns

```

3.3 AbstractComponent module

This modules define the Abstract Component Class

```

class pytanga.components.AbstractComponent.AbstractComponent
    Bases: abc.ABC

    abstract add (component) → None
        This should method add a subComponent.

    abstract getXMLNS ()

    abstract parse (serializer: pytanga.visitors.AbstractVisitor.AbstractVisitor)
        This method should call the parse method for all childrens passing the serializer.

    abstract remove (component) → None
        This should method remove a subComponent.

```

3.4 Config module

Config component.

This module defines the config Component.

```
<config> </config>
```

```

class pytanga.components.config.configComponent
    Bases: pytanga.components.AbstractComponent.AbstractComponent

    add (component) → None
        This should method add a subComponent.

    getXMLNS ()

    is_composite () → bool

    parse (serializer)
        This method should call the parse method for all childrens passing the serializer.

```

remove (*component*) → None

This should method remove a subComponent.

property `xmlns`

4.1 Cisco IOS-XE

4.1.1 ConfigureBGP

```
class pytanga.helpers.Cisco.xe.bgp.ConfigureBGP (asn, aigp_rib_metric=None,  
always_compare_med=None,  
cluster_id=None, deter-  
ministic_med=None, en-  
force_first_as=None, en-  
hanced_error=None,  
fast_external_fallover=None,  
log_neighbor_changes=None,  
maxas_limit=None, max-  
community_limit=None,  
route_map_cache=None, up-  
date_delay=None, router_id=None)
```

Bases: object

```
addAfiSafiNeighbor (afi_safi, address, activate=None, advertisement_interval=None,  
allow_policy=None, allowas_in=None, default_originate=None,  
default_originate_route_map=None, dmzlink_bw=None, maxi-  
mum_prefix_n=None, maximum_prefix_threshold=None, maxi-  
mum_prefix_restart=None, maximum_prefix_warning=None,  
next_hop_self=None, next_hop_self_all=None, next_hop_unchanged=None,  
route_reflector_client=None, send_community=None, send_label=None,  
soft_reconfiguration=None, weight=None)
```

```
addAfiSafiNeighborRouteMap (afi_safi, nei_address, inout, name)
```

```
addAfiSafiNetwork (afi_safi, network, mask=None, route_map=None, backdoor=None)
```

```
addAfi_Safi (afi_name, safi_name)
```

```
addNeighbor (address, remote_as=None, cluster_id=None, description=None, dis-  
able_connected_check=None, ebgp_multihop=None, password=None,  
peer_group=None, shutdown=None, keepalive_interval=None, holdtime=None,  
minimum_neighbor_hold=None, ttl_security=None, update_source=None, ver-  
sion=None)
```

```
configureAfiSafiBGP (afi_safi, advertise_best_external=None, dmzlink_bw=None, sup-  
press_inactive=None, soft_reconfig_backup=None, scan_time=None)
```

```
getBGP ()
```

exception `pytanga.helpers.Cisco.xe.bgp.ConfigureBGPError`
Bases: `Exception`

4.1.2 ConfigurePrefixList

class `pytanga.helpers.Cisco.xe.prefix.ConfigurePrefixList` (*name*, *replace=False*,
step=5)

Bases: `object`

Prefix List configuration helper Class

Parameters

- **name** (*string*) – the prefix list name
- **replace** (*string*) – set to config replace
- **step** (*integer*) – The sequence step for prefix list creation defaults 5

addPrefix (*action*, *network*, *seq=None*, *le=None*, *ge=None*)
Add a prefix to the prefix List

Parameters

- **action** (*string*) – The prefix action should be in [“permit” , “deny”]
- **network** (*string*) – The network
- **seq** (*integer*, *optional*) – The prefix sequence

getPrefixList ()

Returns The prefixList Component

Return type *prefixeslistsComponent*

exception `pytanga.helpers.Cisco.xe.prefix.ConfigurePrefixListError`
Bases: `Exception`

4.2 OpenConfig

4.2.1 CreateIPInterface

`pytanga.helpers.OpenConfig.interfaces.createIPInterface` (*name*, *if_type*, *ip_version*,
address, *prefix_length*,
if_description=None,
if_mtu=None,
enabled=None,
sub_index=0,
sub_desc=None)

VISITORS

5.1 AbstractVisitor module

```
class pytanga.visitors.AbstractVisitor.AbstractVisitor
    Bases: abc.ABC

    abstract parse (leaf)
```

5.2 NETCONFvisitor module

```
class pytanga.visitors.netconfvisitor.NETCONFVisitor
    Bases: pytanga.visitors.AbstractVisitor.AbstractVisitor

    parse (leaf)

    parseComponentData (tag, data, xmlns)

    print (output)

exception pytanga.visitors.netconfvisitor.VisitorError
    Bases: Exception
```


INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

pytanga.components.AbstractComponent, 37
 pytanga.components.Cisco.xe.bgp.addressfamilies, 18
 pytanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast, 17
 pytanga.components.Cisco.xe.bgp.addressFamilyType, 17
 pytanga.components.Cisco.xe.bgp.addressFamilyVR, 18
 pytanga.components.Cisco.xe.bgp.bgp, 19
 pytanga.components.Cisco.xe.bgp.neighbor, 22
 pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap, 23
 pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps, 24
 pytanga.components.Cisco.xe.bgp.neighborDistributeList, 24
 pytanga.components.Cisco.xe.bgp.neighborPrefixList, 24
 pytanga.components.Cisco.xe.bgp.neighborRouteMap, 25
 pytanga.components.Cisco.xe.bgp.network, 25
 pytanga.components.Cisco.xe.bgp.peergroup, 26
 pytanga.components.Cisco.xe.ip.ip, 14
 pytanga.components.Cisco.xe.ip.prefix, 14
 pytanga.components.Cisco.xe.ip.prefixlist, 15
 pytanga.components.Cisco.xe.ip.prefixlists, 15
 pytanga.components.Cisco.xe.ip.routemap, 16
 pytanga.components.Cisco.xe.ip.routemapentry, 16
 pytanga.components.Cisco.xe.native, 13
 pytanga.components.Cisco.xe.ospf.area, 26
 pytanga.components.Cisco.xe.ospf.network, 27
 pytanga.components.Cisco.xe.ospf.ospf, 27
 pytanga.components.Cisco.xe.router, 13
 pytanga.components.Cisco.xr, 28
 pytanga.components.config, 37
 pytanga.components.OpenConfig.interfaces.ethernet, 28
 pytanga.components.OpenConfig.interfaces.interface, 29
 pytanga.components.OpenConfig.interfaces.interfaces, 29
 pytanga.components.OpenConfig.interfaces.oc_ip, 29
 pytanga.components.OpenConfig.interfaces.oc_ipAddress, 30
 pytanga.components.OpenConfig.interfaces.oc_ipAddressList, 30
 pytanga.components.OpenConfig.interfaces.subinterface, 31
 pytanga.components.OpenConfig.interfaces.subinterfaceList, 31
 pytanga.components.OpenConfig.interfaces.switchedVlan, 31
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2, 32
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2Area, 32
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreaList, 33
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interface, 33
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2InterfaceList, 34
 pytanga.components.OpenConfig.routing.ospfv2.ospfv2Network, 34
 pytanga.components.OpenConfig.routing.static.interface, 35
 pytanga.components.OpenConfig.routing.static.nexthop, 35
 pytanga.components.OpenConfig.routing.static.nexthopList, 35

36
pytanga.components.OpenConfig.routing.static.static,
36
pytanga.components.OpenConfig.routing.static.staticroutes,
37
pytanga.helpers.Cisco.xe.bgp, 39
pytanga.helpers.Cisco.xe.prefix, 40
pytanga.helpers.OpenConfig.interfaces,
40
pytanga.visitors.AbstractVisitor, 41
pytanga.visitors.netconfvisitor, 41

INDEX

A

- AbstractComponent (class in py-
tanga.components.AbstractComponent),
37
- AbstractVisitor (class in py-
tanga.visitors.AbstractVisitor), 41
- add () (pytanga.components.AbstractComponent.AbstractComponent
method), 37
- add () (pytanga.components.Cisco.xe.bgp.addressfamilies.addressFamiliesComponent
method), 18
- add () (pytanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast.addressFamilyIPv4UnicastComponent
method), 17
- add () (pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent
method), 17
- add () (pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponent
method), 18
- add () (pytanga.components.Cisco.xe.bgp.bgp.bgpComponent
method), 19
- add () (pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent
method), 20
- add () (pytanga.components.Cisco.xe.bgp.neighbor.neighborComponent
method), 22
- add () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent
method), 23
- add () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent
method), 24
- add () (pytanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent
method), 24
- add () (pytanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent
method), 24
- add () (pytanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponent
method), 25
- add () (pytanga.components.Cisco.xe.bgp.network.networkComponent
method), 25
- add () (pytanga.components.Cisco.xe.bgp.peergroup.peerGroupComponent
method), 26
- add () (pytanga.components.Cisco.xe.ip.ip.ipComponent
method), 14
- add () (pytanga.components.Cisco.xe.ip.prefix.prefixComponent
method), 14
- add () (pytanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent
method), 15
- add () (pytanga.components.Cisco.xe.ip.prefixlists.prefixeslistsComponent
method), 15
- add () (pytanga.components.Cisco.xe.ip.routemap.routemapComponent
method), 16
- add () (pytanga.components.Cisco.xe.ip.routemapentry.routemapentryCom
method), 16
- add () (pytanga.components.Cisco.xe.native.nativeComponent
method), 13
- add () (pytanga.components.Cisco.xe.ospf.area.areaComponent
method), 26
- add () (pytanga.components.Cisco.xe.ospf.network.networkComponent
method), 27
- add () (pytanga.components.Cisco.xe.ospf.ospf.ospfComponent
method), 27
- add () (pytanga.components.Cisco.xe.router.routerComponent
method), 13
- add () (pytanga.components.config.configComponent
method), 37
- add () (pytanga.components.OpenConfig.interfaces.ethernet.ethernetComp
method), 28
- add () (pytanga.components.OpenConfig.interfaces.interface.interfaceCom
method), 29
- add () (pytanga.components.OpenConfig.interfaces.interfaces.interfacesC
method), 29
- add () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponen
method), 29
- add () (pytanga.components.OpenConfig.interfaces.oc_ipAddress.oc_ipAd
method), 30
- add () (pytanga.components.OpenConfig.interfaces.oc_ipAddresses.oc_ip
method), 30
- add () (pytanga.components.OpenConfig.interfaces.subinterface.subinterf
method), 31
- add () (pytanga.components.OpenConfig.interfaces.subinterfaces.subinter
method), 31
- add () (pytanga.components.OpenConfig.interfaces.switchedVlan.switched
method), 31
- add () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2.ospfv2Com
method), 32
- add () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Area.ospfv2
method), 32
- add () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Areas.ospfv
method), 33

add () (*pytanga.components.OpenConfig.routing.ospfv2.ospfv2GlobalComponent* class in *pytanga.components.Cisco.xe.bgp.bgp*), 19
 method), 33

add () (*pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent* method), 34

C

add () (*pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interfaces.ospfv2InterfacesComponent* class in *pytanga.components.config*), 37
 method), 34

add () (*pytanga.components.OpenConfig.routing.static.interfaces.interfacesComponent* (py-
 method), 35 *tanga.helpers.Cisco.xe.bgp.ConfigureBGP*

add () (*pytanga.components.OpenConfig.routing.static.nextHop.nextHopComponent* *ConfigureBGP* (class in *py-
 method*), 35 *tanga.helpers.Cisco.xe.bgp*), 39

add () (*pytanga.components.OpenConfig.routing.static.nextHops.nextHopsComponent* *ConfigureBGPError*, 39
 method), 36

add () (*pytanga.components.OpenConfig.routing.static.staticComponent* *ConfigureBGPError*, 39
 method), 36

add () (*pytanga.components.OpenConfig.routing.static.staticRoutesComponent* *ConfigureBGPError*, 40
 method), 37

addAfi_Safi () (*py-
 tanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

E

addAfiSafiNeighbor () (*py-
 tanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

addAfiSafiNeighborRouteMap () (*py-
 tanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

addAfiSafiNetwork () (*py-
 tanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

addNeighbor () (*py-
 tanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

addPrefix () (*pytanga.helpers.Cisco.xe.prefix.ConfigurePrefixList*
 method), 40

G

getBGP () (*pytanga.helpers.Cisco.xe.bgp.ConfigureBGP*
 method), 39

getPrefixList () (*py-
 tanga.helpers.Cisco.xe.prefix.ConfigurePrefixList*
 method), 40

getXMLNS () (*pytanga.components.AbstractComponent.AbstractComponent*
 method), 37

getXMLNS () (*pytanga.components.Cisco.xe.bgp.addressfamilies.addressFamilyComponent*
 method), 18

getXMLNS () (*pytanga.components.Cisco.xe.bgp.addressFamilyIPv4UnicastComponent*
 method), 17

getXMLNS () (*pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent*
 method), 17

getXMLNS () (*pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponent*
 method), 18

getXMLNS () (*pytanga.components.Cisco.xe.bgp.bgp.bgpComponent*
 method), 19

getXMLNS () (*pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent*
 method), 20

getXMLNS () (*pytanga.components.Cisco.xe.bgp.neighbor.neighborComponent*
 method), 23

getXMLNS () (*pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent*
 method), 23

getXMLNS () (*pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent*
 method), 24

getXMLNS () (*pytanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent*
 method), 24

getXMLNS () (*pytanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent*
 method), 24

B

bgpComponent (class in *py-
 tanga.components.Cisco.xe.bgp.bgp*), 19

getXMLNS () (pytanga.components.Cisco.xe.bgp.neighborRouteMapNeighborRouteMapComponent (py-
method), 25 method), 33

getXMLNS () (pytanga.components.Cisco.xe.bgp.network.networkComponent (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Glo-
method), 25 method), 34

getXMLNS () (pytanga.components.Cisco.xe.bgp.peergroup.peerGroupComponent (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Inte-
method), 26 method), 34

getXMLNS () (pytanga.components.Cisco.xe.ip.ip.ipComponent (pytanga.components.OpenConfig.routing.static.interfaceref-
method), 14 method), 35

getXMLNS () (pytanga.components.Cisco.xe.ip.prefix.prefixComponent (pytanga.components.OpenConfig.routing.static.nextHop.nextHop-
method), 14 method), 35

getXMLNS () (pytanga.components.Cisco.xe.ip.prefixlist.prefixListComponent (pytanga.components.OpenConfig.routing.static.nextHops.nextHops-
method), 15 method), 36

getXMLNS () (pytanga.components.Cisco.xe.ip.prefixlists.prefixListComponent (pytanga.components.OpenConfig.routing.static.static.static-
method), 15 method), 36

getXMLNS () (pytanga.components.Cisco.xe.ip.routemap.routeMapComponent (pytanga.components.OpenConfig.routing.static.static.routes-
method), 16 method), 37

getXMLNS () (pytanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent
method), 16

getXMLNS () (pytanga.components.Cisco.xe.native.nativeComponent (py-
method), 13 tanga.components.OpenConfig.interfaces.interface.interfaceComponent (py-
attribute), 29

getXMLNS () (pytanga.components.Cisco.xe.ospf.area.areaComponent (py-
method), 26 interfaceComponent (class in py-
tanga.components.OpenConfig.interfaces.interface),

getXMLNS () (pytanga.components.Cisco.xe.ospf.network.networkComponent (py-
method), 27 29

getXMLNS () (pytanga.components.Cisco.xe.ospf.ospf.ospfComponent (py-
method), 27 tanga.components.OpenConfig.routing.static.interfaceref),

getXMLNS () (pytanga.components.Cisco.xe.router.routerComponent (py-
method), 13 interfacesComponent (class in py-
tanga.components.OpenConfig.interfaces.interfaces),

getXMLNS () (pytanga.components.config.configComponent (py-
method), 37 29

getXMLNS () (pytanga.components.OpenConfig.interfaces.ethernet.ethernetComponent (class in py-
method), 28 tanga.components.Cisco.xe.ip.ip), 14

getXMLNS () (pytanga.components.OpenConfig.interfaces.interface.interfaceComponent (py-
method), 29 tanga.components.Cisco.xe.bgp.addressfamilies.addressFamiliesComponent (py-
method), 29

getXMLNS () (pytanga.components.OpenConfig.interfaces.interfaces.interfacesComponent (py-
method), 29 is_composite() (py-
tanga.components.Cisco.xe.bgp.addressFamilyIPV4Unicast.addressFamilyIPV4UnicastComponent (py-
method), 17

getXMLNS () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponent (py-
method), 30 tanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent (py-
method), 30

getXMLNS () (pytanga.components.OpenConfig.interfaces.oc_ipAddresses.oc_ipAddressesComponent (py-
method), 30 is_composite() (py-
tanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponent (py-
method), 18

getXMLNS () (pytanga.components.OpenConfig.interfaces.subinterface.subinterfaceComponent (py-
method), 31 tanga.components.Cisco.xe.bgp.bgp.bgpComponent (py-
method), 31

getXMLNS () (pytanga.components.OpenConfig.interfaces.switchedVlan.switchedVlanComponent (py-
method), 31 is_composite() (py-
tanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent (py-
method), 20

getXMLNS () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreaComponent (py-
method), 32 tanga.components.Cisco.xe.bgp.neighbor.neighborComponent (py-
method), 32

getXMLNS () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreasComponent (py-
method), 33

is_composite() tanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent method), 23	(py- is_composite() tanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent method), 37	(py- OpenConfig.interfaces.ethernet.ethernetComponent method), 37
is_composite() tanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent method), 24	(py- is_composite() tanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent method), 28	(py- OpenConfig.interfaces.ethernet.ethernetComponent method), 28
is_composite() tanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent method), 24	(py- is_composite() tanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent method), 29	(py- OpenConfig.interfaces.interface.interfaceComponent method), 29
is_composite() tanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent method), 24	(py- is_composite() tanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent method), 29	(py- OpenConfig.interfaces.interfaces.interfacesComponent method), 29
is_composite() tanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponent method), 25	(py- is_composite() tanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponent method), 29	(py- OpenConfig.interfaces.oc_ip.oc_ipComponent method), 29
is_composite() tanga.components.Cisco.xe.bgp.network.networkComponent method), 25	(py- is_composite() tanga.components.Cisco.xe.bgp.network.networkComponent method), 30	(py- OpenConfig.interfaces.oc_ipAddress.oc_ipAddressComponent method), 30
is_composite() tanga.components.Cisco.xe.bgp.peergroup.peergroupComponent method), 26	(py- is_composite() tanga.components.Cisco.xe.bgp.peergroup.peergroupComponent method), 30	(py- OpenConfig.interfaces.oc_ipAddresses.oc_ipAddressesComponent method), 30
is_composite() tanga.components.Cisco.xe.ip.ip.ipComponent method), 14	(py- is_composite() tanga.components.Cisco.xe.ip.ip.ipComponent method), 31	(py- OpenConfig.interfaces.subinterface.subinterfaceComponent method), 31
is_composite() tanga.components.Cisco.xe.ip.prefix.prefixComponent method), 14	(py- is_composite() tanga.components.Cisco.xe.ip.prefix.prefixComponent method), 31	(py- OpenConfig.interfaces.subinterfaces.subinterfacesComponent method), 31
is_composite() tanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent method), 15	(py- is_composite() tanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent method), 31	(py- OpenConfig.interfaces.switchedVlan.switchedVlanComponent method), 31
is_composite() tanga.components.Cisco.xe.ip.prefixlists.prefixeslistsComponent method), 15	(py- is_composite() tanga.components.Cisco.xe.ip.prefixlists.prefixeslistsComponent method), 32	(py- OpenConfig.routing.ospfv2.ospfv2.ospfv2Component method), 32
is_composite() tanga.components.Cisco.xe.ip.routemap.routemapComponent method), 16	(py- is_composite() tanga.components.Cisco.xe.ip.routemap.routemapComponent method), 32	(py- OpenConfig.routing.ospfv2.ospfv2Area.ospfv2AreaComponent method), 32
is_composite() tanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent method), 16	(py- is_composite() tanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent method), 33	(py- OpenConfig.routing.ospfv2.ospfv2Areas.ospfv2AreasComponent method), 33
is_composite() tanga.components.Cisco.xe.native.nativeComponent method), 13	(py- is_composite() tanga.components.Cisco.xe.native.nativeComponent method), 33	(py- OpenConfig.routing.ospfv2.ospfv2Global.ospfv2GlobalComponent method), 33
is_composite() tanga.components.Cisco.xe.ospf.area.areaComponent method), 26	(py- is_composite() tanga.components.Cisco.xe.ospf.area.areaComponent method), 34	(py- OpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent method), 34
is_composite() tanga.components.Cisco.xe.ospf.network.networkComponent method), 27	(py- is_composite() tanga.components.Cisco.xe.ospf.network.networkComponent method), 34	(py- OpenConfig.routing.ospfv2.ospfv2Interfaces.ospfv2InterfacesComponent method), 34
is_composite() tanga.components.Cisco.xe.ospf.ospf.ospfComponent method), 28	(py- is_composite() tanga.components.Cisco.xe.ospf.ospf.ospfComponent method), 35	(py- OpenConfig.routing.static.interface.ref.interfaceComponent method), 35
is_composite() tanga.components.Cisco.xe.router.routerComponent method), 13	(py- is_composite() tanga.components.Cisco.xe.router.routerComponent method), 35	(py- OpenConfig.routing.static.nextHop.nextHopComponent method), 35

is_composite()	(py- tanga.components.OpenConfig.routing.static.nexthops.nexthopsComponent method), 36	pytanga.components.Cisco.xe.ospf.area, 26
is_composite()	(py- tanga.components.OpenConfig.routing.static.static.staticComponent method), 36	pytanga.components.Cisco.xe.ospf.network, 27
is_composite()	(py- tanga.components.OpenConfig.routing.static.staticroutes.staticroutesComponent method), 37	pytanga.components.Cisco.xe.ospf.ospf, 27
		pytanga.components.Cisco.xe.router, 27
		pytanga.components.Cisco.xr, 28
		pytanga.components.config, 37
		pytanga.components.OpenConfig.interfaces.ethernet, 28
module		
pytanga.components.AbstractComponent,	37	pytanga.components.OpenConfig.interfaces.interf 29
pytanga.components.Cisco.xe.bgp.addressfamily,	18	pytanga.components.OpenConfig.interfaces.interf 29
pytanga.components.Cisco.xe.bgp.addressfamilyipv4,	17	pytanga.components.OpenConfig.interfaces.oc_ip, 29
pytanga.components.Cisco.xe.bgp.addressfamilyipv6,	17	pytanga.components.OpenConfig.interfaces.oc_ipA 30
pytanga.components.Cisco.xe.bgp.addressfamilyvrf,	18	pytanga.components.OpenConfig.interfaces.oc_ipA 30
pytanga.components.Cisco.xe.bgp.bgp,	19	pytanga.components.OpenConfig.interfaces.subint 31
pytanga.components.Cisco.xe.bgp.neighbor,	22	pytanga.components.OpenConfig.interfaces.subint 31
pytanga.components.Cisco.xe.bgp.neighboradvertise,	23	pytanga.components.OpenConfig.interfaces.switch 31
pytanga.components.Cisco.xe.bgp.neighboradvertise,	24	pytanga.components.OpenConfig.routing.ospfv2.os 32
pytanga.components.Cisco.xe.bgp.neighboradvertise,	24	pytanga.components.OpenConfig.routing.ospfv2.os 32
pytanga.components.Cisco.xe.bgp.neighboradvertise,	24	pytanga.components.OpenConfig.routing.ospfv2.os 33
pytanga.components.Cisco.xe.bgp.neighboradvertise,	25	pytanga.components.OpenConfig.routing.ospfv2.os 33
pytanga.components.Cisco.xe.bgp.network,	25	pytanga.components.OpenConfig.routing.ospfv2.os 34
pytanga.components.Cisco.xe.bgp.peergroup,	26	pytanga.components.OpenConfig.routing.ospfv2.os 34
pytanga.components.Cisco.xe.ip.ip,	14	pytanga.components.OpenConfig.routing.static.in 35
pytanga.components.Cisco.xe.ip.prefix,	14	pytanga.components.OpenConfig.routing.static.ne 35
pytanga.components.Cisco.xe.ip.prefixlist,	15	pytanga.components.OpenConfig.routing.static.ne 36
pytanga.components.Cisco.xe.ip.prefixlist,	15	pytanga.components.OpenConfig.routing.static.st 36
pytanga.components.Cisco.xe.ip.routemap,	16	pytanga.components.OpenConfig.routing.static.st 37
pytanga.components.Cisco.xe.ip.routemap,	16	pytanga.helpers.Cisco.xe.bgp, 39
		pytanga.helpers.Cisco.xe.prefix, 40
pytanga.components.Cisco.xe.native,	13	pytanga.helpers.OpenConfig.interfaces, 40

pytanga.visitors.AbstractVisitor, 41
 pytanga.visitors.netconfvisitor, 41

N

nativeComponent (class in py-
 tanga.components.Cisco.xe.native), 13
 neighborAdvertiseMapComponent (class in py-
 tanga.components.Cisco.xe.bgp.neighborAdvertiseMap),
 23
 neighborAdvertiseMapsComponent (class in py-
 tanga.components.Cisco.xe.bgp.neighborAdvertiseMaps),
 24
 neighborComponent (class in py-
 tanga.components.Cisco.xe.bgp.neighbor),
 22
 neighborDistributeListComponent
 (class in py-
 tanga.components.Cisco.xe.bgp.neighborDistributeList),
 24

neighborPrefixListComponent (class in py-
 tanga.components.Cisco.xe.bgp.neighborPrefixList),
 24

neighborRouteMapComponent (class in py-
 tanga.components.Cisco.xe.bgp.neighborRouteMap),
 25

neighborSyntaxError, 23

NETCONFVisitor (class in py-
 tanga.visitors.netconfvisitor), 41

networkComponent (class in py-
 tanga.components.Cisco.xe.bgp.network),
 25

networkComponent (class in py-
 tanga.components.Cisco.xe.ospf.network),
 27

nexthopComponent (class in py-
 tanga.components.OpenConfig.routing.static.nexthop),
 35

nexthopsComponent (class in py-
 tanga.components.OpenConfig.routing.static.nexthops),
 36

O

oc_ipAddressComponent (class in py-
 tanga.components.OpenConfig.interfaces.oc_ipAddress),
 30

oc_ipAddressesComponent (class in py-
 tanga.components.OpenConfig.interfaces.oc_ipAddresses),
 30

oc_ipComponent (class in py-
 tanga.components.OpenConfig.interfaces.oc_ip),
 29

ospfComponent (class in py-
 tanga.components.Cisco.xe.ospf.ospf), 27

ospfv2AreaComponent (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2Area),
 32

ospfv2AreasComponent (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2Areas),
 33

ospfv2Component (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2),
 32

ospfv2GlobalComponent (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2Global),
 33

ospfv2InterfaceComponent (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2Interface),
 34

ospfv2InterfacesComponent (class in py-
 tanga.components.OpenConfig.routing.ospfv2.ospfv2Interfaces),
 34

P

parent () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComp
 property), 29

parent () (pytanga.components.OpenConfig.interfaces.oc_ipAddress.oc_
 property), 30

parent () (pytanga.components.OpenConfig.interfaces.oc_ipAddresses.o
 property), 30

parse () (pytanga.components.AbstractComponent.AbstractComponent
 method), 37

parse () (pytanga.components.Cisco.xe.bgp.addressfamilies.addressFami
 method), 18

parse () (pytanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast.a
 method), 17

parse () (pytanga.components.Cisco.xe.bgp.addressFamilyType.addressF
 method), 17

parse () (pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressF
 method), 18

parse () (pytanga.components.Cisco.xe.bgp.bgp.bgpComponent
 method), 19

parse () (pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent
 method), 20

parse () (pytanga.components.Cisco.xe.bgp.neighbor.neighborComponen
 method), 23

parse () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neigh
 method), 23

parse () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neigh
 method), 24

parse () (pytanga.components.Cisco.xe.bgp.neighborDistributeList.neigh
 method), 24

parse () (pytanga.components.Cisco.xe.bgp.neighborPrefixList.neighbor
 method), 24

parse () (pytanga.components.Cisco.xe.bgp.neighborRouteMap.neighbor
 method), 25

parse () (pytanga.components.Cisco.xe.bgp.network.networkComponent
 method), 25

parse () (pytanga.components.Cisco.xe.bgp.peergroup.peergroupComponent (pytanga.components.OpenConfig.routing.ospfv2.ospfv2InterfaceComponent), 26
 parse () (pytanga.components.Cisco.xe.ip.ipComponent (pytanga.components.OpenConfig.routing.static.interfaceref.interfaceComponent), 14
 parse () (pytanga.components.Cisco.xe.ip.prefix.prefixComponent (pytanga.components.OpenConfig.routing.static.nextthop.nextthopComponent), 14
 parse () (pytanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent (pytanga.components.OpenConfig.routing.static.nextthops.nextthopsComponent), 15
 parse () (pytanga.components.Cisco.xe.ip.prefixlists.prefixlistsComponent (pytanga.components.OpenConfig.routing.static.static.staticComponent), 15
 parse () (pytanga.components.Cisco.xe.ip.routemap.routemapComponent (pytanga.components.OpenConfig.routing.static.staticroutes.staticroutesComponent), 16
 parse () (pytanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent (pytanga.components.OpenConfig.routing.static.staticroutes.staticroutesComponent), 16
 parse () (pytanga.components.Cisco.xe.native.nativeComponent (pytanga.visitors.netconfvisitor.NETCONFVisitor), 13
 parse () (pytanga.components.Cisco.xe.ospf.area.areaComponent (pytanga.components.OpenConfig.routing.ospf.area.areaComponentData () (pytanga.visitors.netconfvisitor.NETCONFVisitor), 26
 parse () (pytanga.components.Cisco.xe.ospf.network.networkComponent (pytanga.components.OpenConfig.routing.ospf.network.networkComponent), 27
 parse () (pytanga.components.Cisco.xe.ospf.ospf.ospfComponent (pytanga.components.Cisco.xe.bgp.peergroup), 28
 parse () (pytanga.components.Cisco.xe.router.routerComponent (pytanga.components.Cisco.xe.ip.prefix), 13
 parse () (pytanga.components.config.configComponent (pytanga.components.Cisco.xe.ip.prefixlists), 37
 parse () (pytanga.components.OpenConfig.interfaces.ethernet.ethernetComponent (pytanga.components.Cisco.xe.ip.prefixlist), 28
 parse () (pytanga.components.OpenConfig.interfaces.interface.interfaceComponent (pytanga.components.Cisco.xe.ip.prefixlist), 29
 parse () (pytanga.components.OpenConfig.interfaces.interfaces.interfacesComponent (pytanga.components.Cisco.xe.ip.prefixlist), 29
 parse () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponent (pytanga.visitors.netconfvisitor.NETCONFVisitor), 29
 parse () (pytanga.components.OpenConfig.interfaces.oc_ip_addresses.oc_ip_addressesComponent (pytanga.components.OpenConfig.interfaces.oc_ip_addresses.oc_ip_addressesComponent), 30
 parse () (pytanga.components.OpenConfig.interfaces.oc_ip_addresses.oc_ip_addressesComponent (pytanga.components.Cisco.xe.bgp.addressfamilies), 30
 parse () (pytanga.components.OpenConfig.interfaces.subinterfaces.subinterfaceComponent (pytanga.components.Cisco.xe.bgp.addressFamilyIPv4), 31
 parse () (pytanga.components.OpenConfig.interfaces.subinterfaces.subinterfacesComponent (pytanga.components.Cisco.xe.bgp.addressFamilyType), 31
 parse () (pytanga.components.OpenConfig.interfaces.switchedVlans.switchedVlanComponent (pytanga.components.Cisco.xe.bgp.addressFamilyVRF), 31
 parse () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Component (pytanga.components.Cisco.xe.bgp.bgp), 32
 parse () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreaComponent (pytanga.components.Cisco.xe.bgp.neighbor), 32
 parse () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreasComponent (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap), 33
 parse () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2GlobalComponent (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap), 33
 parse () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2InterfaceComponent (pytanga.components.Cisco.xe.bgp.neighborDistributeList), 34

module, 24
 pytanga.components.Cisco.xe.bgp.neighboradvertisemap module, 25
 pytanga.components.Cisco.xe.bgp.network module, 25
 pytanga.components.Cisco.xe.bgp.peergroup module, 26
 pytanga.components.Cisco.xe.ip.ip module, 14
 pytanga.components.Cisco.xe.ip.prefix module, 14
 pytanga.components.Cisco.xe.ip.prefixlist module, 15
 pytanga.components.Cisco.xe.ip.prefixlist module, 15
 pytanga.components.Cisco.xe.ip.routemap module, 16
 pytanga.components.Cisco.xe.ip.routemap module, 16
 pytanga.components.Cisco.xe.native module, 13
 pytanga.components.Cisco.xe.ospf.area module, 26
 pytanga.components.Cisco.xe.ospf.network module, 27
 pytanga.components.Cisco.xe.ospf.ospf module, 27
 pytanga.components.Cisco.xe.router module, 13
 pytanga.components.Cisco.xr module, 28
 pytanga.components.config module, 37
 pytanga.components.OpenConfig.interfaces module, 28
 pytanga.components.OpenConfig.interfaces module, 29
 pytanga.components.OpenConfig.interfaces module, 29
 pytanga.components.OpenConfig.interfaces module, 29
 pytanga.components.OpenConfig.interfaces module, 30
 pytanga.components.OpenConfig.interfaces module, 30
 pytanga.components.OpenConfig.interfaces module, 31
 pytanga.components.OpenConfig.interfaces module, 31
 pytanga.components.OpenConfig.interfaces module, 31
 pytanga.components.OpenConfig.interfaces module, 31
 pytanga.components.OpenConfig.routing.ospfv2 module, 32
 pytanga.components.OpenConfig.routing.ospfv2 module, 32
 pytanga.components.OpenConfig.routing.ospfv2 module, 33
 pytanga.components.OpenConfig.routing.ospfv2 module, 33
 pytanga.components.OpenConfig.routing.ospfv2 module, 34
 pytanga.components.OpenConfig.routing.ospfv2 module, 34
 pytanga.components.OpenConfig.routing.static.interfaces module, 35
 pytanga.components.OpenConfig.routing.static.next hop module, 35
 pytanga.components.OpenConfig.routing.static.next hop module, 36
 pytanga.components.OpenConfig.routing.static.static module, 36
 pytanga.components.OpenConfig.routing.static.static module, 37
 pytanga.helpers.Cisco.xe.bgp module, 39
 pytanga.helpers.Cisco.xe.prefix module, 40
 pytanga.helpers.OpenConfig.interfaces module, 40
 pytanga.visitors.AbstractVisitor module, 41
 pytanga.visitors.netconfvisitor module, 41

R

remove () (pytanga.components.AbstractComponent.AbstractComponent method), 37
 remove () (pytanga.components.Cisco.xe.bgp.addressfamilies.addressFamily method), 18
 remove () (pytanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast method), 17
 remove () (pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyType method), 17
 remove () (pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRF method), 18
 remove () (pytanga.components.Cisco.xe.bgp.bgp.bgpComponent method), 19
 remove () (pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent method), 20
 remove () (pytanga.components.Cisco.xe.bgp.neighbor.neighborComponent method), 23
 remove () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMap method), 23
 remove () (pytanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMaps method), 24
 remove () (pytanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeList method), 24

remove () (pytanga.components.Cisco.xe.bgp.neighborPrefixList.remove (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Areas.
method), 24 method), 33

remove () (pytanga.components.Cisco.xe.bgp.neighborRouteMap.remove (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Global
method), 25 method), 33

remove () (pytanga.components.Cisco.xe.bgp.network.networkComponent.remove (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interfa
method), 25 method), 34

remove () (pytanga.components.Cisco.xe.bgp.peergroup.peerGroupComponent.remove (pytanga.components.OpenConfig.routing.ospfv2.ospfv2Interfa
method), 26 method), 34

remove () (pytanga.components.Cisco.xe.ip.ip.ipComponent.remove (pytanga.components.OpenConfig.routing.static.interfaceref.in
method), 14 method), 35

remove () (pytanga.components.Cisco.xe.ip.prefix.prefixComponent.remove (pytanga.components.OpenConfig.routing.static.nextHop.nextHo
method), 14 method), 35

remove () (pytanga.components.Cisco.xe.ip.prefixlist.prefixListComponent.remove (pytanga.components.OpenConfig.routing.static.nextHops.nextH
method), 15 method), 36

remove () (pytanga.components.Cisco.xe.ip.prefixlists.prefixListComponent.remove (pytanga.components.OpenConfig.routing.static.static.staticCo
method), 15 method), 36

remove () (pytanga.components.Cisco.xe.ip.routemap.routemapComponent.remove (pytanga.components.OpenConfig.routing.static.static.routes.sta
method), 16 method), 37

remove () (pytanga.components.Cisco.xe.ip.routemapentry.routemapEntryComponent (class in py-
method), 16 tanga.components.Cisco.xe.ip.routemap),

remove () (pytanga.components.Cisco.xe.native.nativeComponent (class in py-
method), 13 routemapentryComponent (class in py-
method), 16 tanga.components.Cisco.xe.ip.routemapentry),

remove () (pytanga.components.Cisco.xe.ospf.area.areaComponent (class in py-
method), 27 taxError, 16

remove () (pytanga.components.Cisco.xe.ospf.network.networkComponent (class in py-
method), 27 routerComponent (class in py-
method), 28 tanga.components.Cisco.xe.router), 13

remove () (pytanga.components.Cisco.xe.router.routerComponent (class in py-
method), 13 setAttributes () (py-
method), 37 tanga.components.Cisco.xe.bgp.addressfamilies.addressFamiliesComponent (class in py-
method), 18

remove () (pytanga.components.OpenConfig.interfaces.ethernet.ethernetComponent (class in py-
method), 28 tanga.components.Cisco.xe.bgp.addressFamilyIPV4Unicast.addressFamilyIPV4UnicastComponent (class in py-
method), 17

remove () (pytanga.components.OpenConfig.interfaces.interface.interfaceComponent (class in py-
method), 29 setAttributes () (py-
method), 17 tanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent (class in py-
method), 17

remove () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponent (class in py-
method), 30 setAttributes () (py-
method), 30 tanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponent (class in py-
method), 19

remove () (pytanga.components.OpenConfig.interfaces.oc_ipAddresses.oc_ipAddressesComponent (class in py-
method), 30 setAttributes () (py-
method), 19 tanga.components.Cisco.xe.bgp.bgp.bgpComponent (class in py-
method), 19

remove () (pytanga.components.OpenConfig.interfaces.subinterface.subinterfaceComponent (class in py-
method), 31 setAttributes () (py-
method), 19 tanga.components.Cisco.xe.bgp.bgp.bgpConfigComponent (class in py-
method), 19

remove () (pytanga.components.OpenConfig.interfaces.subinterfaces.subinterfacesComponent (class in py-
method), 31 setAttributes () (py-
method), 23 tanga.components.Cisco.xe.bgp.neighbor.neighborComponent (class in py-
method), 23

remove () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreaComponent (class in py-
method), 32 setAttributes () (py-
method), 23 tanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent (class in py-
method), 23

remove () (pytanga.components.OpenConfig.routing.ospfv2.ospfv2AreaComponent (class in py-
method), 32 setAttributes () (py-
method), 23 tanga.components.Cisco.xe.bgp.neighborAdvertiseMap.neighborAdvertiseMapComponent (class in py-
method), 23

setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.neighborAdvertiseMaps.neighborAdvertiseMapsComponent.interfaces.oc_ipAddress.oc_ipAddressComponent), 24 method), 30
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponent.interfaces.subinterface.subinterfaceComponent), 24 method), 31
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponent.interfaces.switchedVlan.switchedVlanComponent), 25 method), 32
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponent.interfaces.routing.ospfv2.ospfv2Area.ospfv2AreaComponent), 25 method), 32
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.network.networkComponent.tanga.components.OpenConfig.routing.ospfv2.ospfv2Global.ospfv2GlobalComponent), 25 method), 33
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.bgp.peergroup.peergroupComponent.tanga.components.OpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent), 26 method), 34
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.ip.ip.ipComponent tanga.components.OpenConfig.routing.static.interfaceRef.interfaceRefComponent), 14 method), 35
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.ip.prefix.prefixComponent tanga.components.OpenConfig.routing.static.nextHop.nextHopComponent), 14 method), 35
 setAttributes () (py- setAttributes () (py-
 tanga.components.Cisco.xe.ip.prefixlist.prefixlistComponent.tanga.components.OpenConfig.routing.static.static.staticComponent), 15 method), 36
 setAttributes () (py- staticComponent (class in py-
 tanga.components.Cisco.xe.ip.prefixlists.prefixeslistsComponent.tanga.components.OpenConfig.routing.static.static), 15 method), 36
 setAttributes () (py- staticroutesComponent (class in py-
 tanga.components.Cisco.xe.ip.routemap.routemapComponent.tanga.components.OpenConfig.routing.static.staticroutes), 16 method), 37
 setAttributes () (py- subinterfaceComponent (class in py-
 tanga.components.Cisco.xe.ip.routemapentry.routemapentryComponent.tanga.components.OpenConfig.interfaces.subinterface), 16 method), 31
 setAttributes () (py- subinterfacesComponent (class in py-
 tanga.components.Cisco.xe.ospf.area.areaComponent tanga.components.OpenConfig.interfaces.subinterfaces), 27 method), 31
 setAttributes () (py- switchedVlanComponent (class in py-
 tanga.components.Cisco.xe.ospf.network.networkComponent.tanga.components.OpenConfig.interfaces.switchedVlan), 27 method), 31
 setAttributes () (py- **V**
 tanga.components.Cisco.xe.ospf.ospf.ospfComponent method), 28 VisitorError, 41
 setAttributes () (py- **X**
 tanga.components.Cisco.xe.router.routerComponent method), 13
 setAttributes () (py-
 tanga.components.OpenConfig.interfaces.ethernet.ethernetComponent.tanga.components.Cisco.xe.bgp.addressFamilies.addressFamiliesComponent), 28 method), 18
 setAttributes () (py-
 tanga.components.OpenConfig.interfaces.interface.interfaceComponent.tanga.components.Cisco.xe.bgp.addressFamilyIPv4Unicast.addressFamilyIPv4UnicastComponent), 29 method), 17
 setAttributes () (py- xmlns () (pytanga.components.Cisco.xe.bgp.addressFamilyType.addressFamilyTypeComponent), 29 method), 17

xmlns () (pytanga.components.Cisco.xe.bgp.addressFamilyVRF.addressFamilyVRFComponentOpenConfig.interfaces.oc_ipAddress.oc_ip
 property), 18
 xmlns () (pytanga.components.Cisco.xe.bgp.bgp.bgpComponentOpenConfig.interfaces.oc_ipAddresses.oc_ipAddressesComponent
 property), 19
 xmlns () (pytanga.components.Cisco.xe.bgp.bgp.bgpConfigComponentOpenConfig.interfaces.subinterface.subinterfaceComponent
 property), 20
 xmlns () (pytanga.components.Cisco.xe.bgp.neighbor.neighborComponentOpenConfig.interfaces.subinterfaces.subinterfacesComponent
 property), 23
 xmlns () (pytanga.components.Cisco.xe.bgp.neighborAdvertisementMap.neighborAdvertisementMapComponentOpenConfig.interfaces.switchedVlan.switchedVlanComponent
 property), 23
 xmlns () (pytanga.components.Cisco.xe.bgp.neighborAdvertisementMap.neighborAdvertisementMapComponentOpenConfig.routing.ospfv2.ospfv2.ospfv2Component
 property), 24
 xmlns () (pytanga.components.Cisco.xe.bgp.neighborDistributeList.neighborDistributeListComponentOpenConfig.routing.ospfv2.ospfv2Area.ospfv2AreaComponent
 property), 24
 xmlns () (pytanga.components.Cisco.xe.bgp.neighborPrefixList.neighborPrefixListComponentOpenConfig.routing.ospfv2.ospfv2Areas.ospfv2AreasComponent
 property), 25
 xmlns () (pytanga.components.Cisco.xe.bgp.neighborRouteMap.neighborRouteMapComponentOpenConfig.routing.ospfv2.ospfv2Global.ospfv2GlobalComponent
 property), 25
 xmlns () (pytanga.components.Cisco.xe.bgp.network.networkComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 25
 xmlns () (pytanga.components.Cisco.xe.bgp.peergroup.peergroupComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 26
 xmlns () (pytanga.components.Cisco.xe.ip.ip.ipComponentOpenConfig.routing.static.interfaceref.interfaceComponent
 property), 14
 xmlns () (pytanga.components.Cisco.xe.ip.prefix.prefixComponentOpenConfig.routing.static.nextHop.nextHopComponent
 property), 14
 xmlns () (pytanga.components.Cisco.xe.ip.prefixlist.prefixListComponentOpenConfig.routing.static.nextHops.nextHopsComponent
 property), 15
 xmlns () (pytanga.components.Cisco.xe.ip.prefixlists.prefixlistsComponentOpenConfig.routing.static.static.staticComponent
 property), 15
 xmlns () (pytanga.components.Cisco.xe.ip.routemap.routemapComponentOpenConfig.routing.static.staticroutes.staticroutesComponent
 property), 16
 xmlns () (pytanga.components.Cisco.xe.ip.routemapentry.routemapentryComponentOpenConfig.routing.static.staticroutes.staticroutesComponent
 property), 16
 xmlns () (pytanga.components.Cisco.xe.native.nativeComponentOpenConfig.routing.static.staticroutes.staticroutesComponent
 property), 13
 xmlns () (pytanga.components.Cisco.xe.ospf.area.areaComponentOpenConfig.routing.ospfv2.ospfv2Area.ospfv2AreaComponent
 property), 27
 xmlns () (pytanga.components.Cisco.xe.ospf.network.networkComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 27
 xmlns () (pytanga.components.Cisco.xe.ospf.ospf.ospfComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 28
 xmlns () (pytanga.components.Cisco.xe.router.routerComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 13
 xmlns () (pytanga.components.config.configComponentOpenConfig.routing.ospfv2.ospfv2Interface.ospfv2InterfaceComponent
 property), 38
 xmlns () (pytanga.components.OpenConfig.interfaces.ethernet.ethernetComponentOpenConfig.interfaces.ethernet.ethernetComponent
 property), 28
 xmlns () (pytanga.components.OpenConfig.interfaces.interface.interfaceComponentOpenConfig.interfaces.ethernet.ethernetComponent
 property), 29
 xmlns () (pytanga.components.OpenConfig.interfaces.interfaces.interfacesComponentOpenConfig.interfaces.ethernet.ethernetComponent
 property), 29
 xmlns () (pytanga.components.OpenConfig.interfaces.oc_ip.oc_ipComponentOpenConfig.interfaces.ethernet.ethernetComponent
 property), 30