
etherweaver Documentation

Release 0.2.3

Grayson Head

May 05, 2019

Contents:

1	Install Instructions	1
1.1	Development Install	1
2	Getting Started	3
2.1	Config Structure	3
2.2	Nodes	4
2.3	Commands	5
2.4	Examples	6
3	Configuration Elements	13
3.1	Nodes	13
3.2	List Expansion	17
3.3	Interface Profiles	17
3.4	Iterator Keys	18
4	Writing Plugins	21
4.1	Plugin Class Structure	21
5	License	29
6	Indices and tables	43

CHAPTER 1

Install Instructions

1.1 Development Install

1.1.1 Install using pip

```
pip3 install etherweaver
```

1.1.2 Install from Git repo

```
pip3 install git+https://github.com/graysonhead/etherweaver.git
```


2.1 Config Structure

2.1.1 Top Level objects

Etherweaver's can consist of two sections, of which Fabrics are optional:

Listing 1: config.yaml

```
fabrics: # A list of all fabrics
appliances: # A list of all hardware
```

Fabrics represent networks, or collections of network devices with inherited settings.

An appliance represents a network operating system to be configured, and it inherits settings (as well as defining it's own).

2.1.2 An Example Config

Listing 2: config.yaml

```
fabrics:
  network1:
    vlan:
      4-10
    connections:
      ssh:
        username: user
        password: password!

    distribution:
      fabric: network1
```

(continues on next page)

(continued from previous page)

```
interfaces:
  1G:
    1-22:
      untagged_vlan: 4
      tagged_vlans: 5-7
    23-24:
      tagged_vlans: 4-10
appliances:
  distsw1:
    fabric: distribution
    plugin_package: cumulus
    connections:
      ssh:
        hostname: 10.5.5.33
```

The inheritance structure flows in this manner:

Fabric -> Child Fabric*n -> appliances.

A node specified twice will take the value of the last object to specify it in the structure.

2.2 Nodes

Nodes represent a dictionary structure that allow you to access, modify, or remove the current state of an appliance without writing any YAML. A node is represented by it's path in the dict (which is the same as the config.yaml)

For instance, here are some valid nodes:

- protocols.dns.nameservers
- interfaces.1G.1
- hostname

Nodes have different commands depending on their type. For instance single value nodes (such as hostname) generally have three commands:

- hostname.get
- hostname.set
- hostname.del

List nodes will often have more command types:

- protocols.ntp.client.servers.add: Adds a server or servers
- protocols.ntp.client.servers.get: Gets a list of all servers
- protocols.ntp.client.servers.set: Overwrites the server list with a new list
- protocols.ntp.client.servers.del: If a value is specified, deletes the value, otherwise deletes all values

Additionally, there are a few meta nodes such as 'state'. State is likely the one you will use the most, and it has one function:

- state.apply: Applies the current config.yaml in an interactive manner

In addition there are also two additional state nodes with the following commands:

- cstate.get: Fetch current state of the appliance

- `dstate.get`: Fetch desired state of the appliance (based on `config.yaml`)
- `dstate.apply`: Non-interactively apply the desired state without confirmation

2.3 Commands

Commands follow a simple syntax:

`netweaver 'role|*' node value --yaml=config.yaml`

The YAML state can be applied to every appliance in the infrastructure file by running the following:

```
netweaver.py '*' state.apply --yaml=config.yaml
sw1: []
sw2: []
```

The brackets will contain a list of any commands run in order to bring the switches in alignment with the current state.

You can view the current state of all appliances in the environment using the following command:

```
netweaver.py 'sw1' cstate.get --yaml=config.yaml
sw1:
  hostname: spine1.net.testco.org
  interfaces:
    100G: {}
    10G: {}
    1G:
      '1':
        ip:
          address: []
          tagged_vlans: [2, 3, 4]
          untagged_vlan: '7'
    ...
```

```
etherweaver.py '*' protocols.ntp.client.servers.get --yaml=config.yaml
sw1: [pool.ntp.org, 0.cumulusnetworks.pool.ntp.org, 1.cumulusnetworks.pool.ntp.org,
2.cumulusnetworks.pool.ntp.org]
sw2: [0.cumulusnetworks.pool.ntp.org, 1.cumulusnetworks.pool.ntp.org, 2.
↪cumulusnetworks.pool.ntp.org,
pool.ntp.org]
```

```
netweaver.py 'sw1' hostname.set 'spine2' --yaml=config.yaml
net add hostname spine2
```

Note: Not all appliance plugins can implement all nodes due to hardware limitations, accessing any unsupported node will result in a `NotImplemented` or `NotSupported` error.

2.4 Examples

2.4.1 A Simple Example

Lets imagine that you are tasked with deploying a switch fabric for a small branch office, with two switches total. Based on the needs of the office, you determine that you need two 24 port switches, which are to be configured using the following rules:

- VLANs for Employees, VOIP phones, servers, Public Wireless, and Management interfaces
- Ports 1-20 on both switches will be for employee usage, and will need the Employee VLAN untagged and the VOIP vlan tagged for phone passthrough
- Ports 20-22 on both switches are reserved for Wireless access points, and need to be untagged on the Management VLAN for AP administration, and tagged on Employee and Public
- Port 24 will be the trunk between switches

From the system you are running etherweaver from, copy your public ssh keys to the switches (For switches that cannot do this, you can use a username and password, but you still need to accept the public ssh key of the system on your machine to prevent man in the middle attacks.) Then you write an etherweaver state file, `simple_example.yaml`:

```
fabrics:
  network:
    vlans:
      10:
        Name: Employee
      11:
        Name: Servers
      12:
        Name: VOIP
      20:
        Name: Public Wireless
      254:
        Name: Management
    port_profiles:
      OfficePorts:
        untagged_vlan: 10
        tagged_vlans: [12]
      WirelessAccessPoint:
        untagged_vlan: 254
        tagged_vlans: [10, 20]
      Trunk:
        tagged_vlans: [10, 12, 20, 254]
  dist_switches:
    fabric: network
    interfaces:
      1G:
        1-20:
          profile: OfficePorts
        20-22:
          profile: WirelessAccessPoint
        24:
          profile: Trunk
  appliances:
    dist1:
      fabric: dist_switches
      plugin_package: cumulus
```

(continues on next page)

(continued from previous page)

```

connections:
  ssh:
    hostname: 192.168.122.254
    username: cumulus
    port: 22
  hostname: dist1
dist2:
  fabric: dist_switches
  plugin_package: cumulus
  connections:
    ssh:
      hostname: 192.168.122.171
      username: cumulus
      port: 22
    hostname: dist2

```

Then, run:

```

etherweaver '*' state.apply --yaml=simple_example.yaml
If you continue, the following changes will be applied:
#####
{  'dist1': [  'net add hostname dist1',
               'net add bridge bridge vids 10-12,20,254',
               'net add interface swp1 bridge vids 12',
               'net add interface swp1 bridge pvid 10',
               'net add interface swp2 bridge vids 12',
               'net add interface swp2 bridge pvid 10',
               'net add interface swp3 bridge vids 12',
               'net add interface swp3 bridge pvid 10',
               'net add interface swp4 bridge vids 12',
               'net add interface swp4 bridge pvid 10',
               'net add interface swp5 bridge vids 12',
               'net add interface swp5 bridge pvid 10',
               'net add interface swp6 bridge vids 12',
               'net add interface swp6 bridge pvid 10',
               'net add interface swp7 bridge vids 12',
               'net add interface swp7 bridge pvid 10',
               'net add interface swp8 bridge vids 12',
               'net add interface swp8 bridge pvid 10',
               'net add interface swp9 bridge vids 12',
               'net add interface swp9 bridge pvid 10',
               'net add interface swp10 bridge vids 12',
               'net add interface swp10 bridge pvid 10',
               'net add interface swp11 bridge vids 12',
               'net add interface swp11 bridge pvid 10',
               'net add interface swp12 bridge vids 12',
               'net add interface swp12 bridge pvid 10',
               'net add interface swp13 bridge vids 12',
               'net add interface swp13 bridge pvid 10',
               'net add interface swp14 bridge vids 12',
               'net add interface swp14 bridge pvid 10',
               'net add interface swp15 bridge vids 12',
               'net add interface swp15 bridge pvid 10',
               'net add interface swp16 bridge vids 12',
               'net add interface swp16 bridge pvid 10',
               'net add interface swp17 bridge vids 12',
               'net add interface swp17 bridge pvid 10',

```

(continues on next page)

(continued from previous page)

```

'net add interface swp18 bridge vids 12',
'net add interface swp18 bridge pvid 10',
'net add interface swp19 bridge vids 12',
'net add interface swp19 bridge pvid 10',
'net add interface swp20 bridge vids 10,20',
'net add interface swp20 bridge pvid 254',
'net add interface swp21 bridge vids 10,20',
'net add interface swp21 bridge pvid 254',
'net add interface swp22 bridge vids 10,20',
'net add interface swp22 bridge pvid 254',
'net add interface swp24 bridge vids 10,12,20,254'],
'dist2': [
'net add hostname dist2',
'net add bridge bridge vids 10-12,20,254',
'net add interface swp1 bridge vids 12',
'net add interface swp1 bridge pvid 10',
'net add interface swp2 bridge vids 12',
'net add interface swp2 bridge pvid 10',
'net add interface swp3 bridge vids 12',
'net add interface swp3 bridge pvid 10',
'net add interface swp4 bridge vids 12',
'net add interface swp4 bridge pvid 10',
'net add interface swp5 bridge vids 12',
'net add interface swp5 bridge pvid 10',
'net add interface swp6 bridge vids 12',
'net add interface swp6 bridge pvid 10',
'net add interface swp7 bridge vids 12',
'net add interface swp7 bridge pvid 10',
'net add interface swp8 bridge vids 12',
'net add interface swp8 bridge pvid 10',
'net add interface swp9 bridge vids 12',
'net add interface swp9 bridge pvid 10',
'net add interface swp10 bridge vids 12',
'net add interface swp10 bridge pvid 10',
'net add interface swp11 bridge vids 12',
'net add interface swp11 bridge pvid 10',
'net add interface swp12 bridge vids 12',
'net add interface swp12 bridge pvid 10',
'net add interface swp13 bridge vids 12',
'net add interface swp13 bridge pvid 10',
'net add interface swp14 bridge vids 12',
'net add interface swp14 bridge pvid 10',
'net add interface swp15 bridge vids 12',
'net add interface swp15 bridge pvid 10',
'net add interface swp16 bridge vids 12',
'net add interface swp16 bridge pvid 10',
'net add interface swp17 bridge vids 12',
'net add interface swp17 bridge pvid 10',
'net add interface swp18 bridge vids 12',
'net add interface swp18 bridge pvid 10',
'net add interface swp19 bridge vids 12',
'net add interface swp19 bridge pvid 10',
'net add interface swp20 bridge vids 10,20',
'net add interface swp20 bridge pvid 254',
'net add interface swp21 bridge vids 10,20',
'net add interface swp21 bridge pvid 254',
'net add interface swp22 bridge vids 10,20',
'net add interface swp22 bridge pvid 254',

```

(continues on next page)

(continued from previous page)

```
'net add interface swp24 bridge vids 10,12,20,254']}]
Do you want to continue? y/[n]y
dist1: 100%|| 47/47 [00:04<00:00, 10.45it/s]
dist2: 100%|| 47/47 [00:08<00:00, 12.90it/s]
Run Complete
```

Now your switches are configured correctly, subsequent runs won't do anything because the current state and desired state match.

To complicate matters, a developer now needs to have a development server at his desk. Unfortunately, his port is right in the middle of our 10-20 range, at port 11 on dist1. Not to worry though, we can place a config statement at any lower inheritance level to override the port range. All we need to do is add an interface definition for the port in question, and define profile to false in order to stop inheritance. Now our state file looks like this:

```
fabrics:
  network:
    vlans:
      10:
        Name: Employee
      11:
        Name: Servers
      12:
        Name: VOIP
      20:
        Name: Public Wireless
      254:
        Name: Management
    port_profiles:
      OfficePorts:
        untagged_vlan: 10
        tagged_vlans: [12]
      WirelessAccessPoint:
        untagged_vlan: 254
        tagged_vlans: [10, 20]
      Trunk:
        tagged_vlans: [10, 12, 20, 254]
    dist_switches:
      fabric: network
      interfaces:
        1G:
          1-20:
            profile: OfficePorts
          20-22:
            profile: WirelessAccessPoint
          24:
            profile: Trunk
  appliances:
    dist1:
      interfaces:
        1G:
          11:
            profile: false
            untagged_vlan: 11
      fabric: dist_switches
      plugin_package: cumulus
      connections:
        ssh:
```

(continues on next page)

(continued from previous page)

```

    hostname: 192.168.122.254
    username: cumulus
    port: 22
  hostname: dist1
dist2:
  fabric: dist_switches
  plugin_package: cumulus
  connections:
    ssh:
      hostname: 192.168.122.171
      username: cumulus
      port: 22
    hostname: dist2

```

And running the program gives us the following output:

```

$etherweaver '*' state.apply --yaml=docs/source/ExampleConfigs/simple_examplev2.yaml
If you continue, the following changes will be applied:
#####
{'dist1': ['net add interface swp11 bridge pvid 11']}
Do you want to continue? y/[n]y
dist1: 100%|| 1/1 [00:00<00:00, 1.08it/s]
Run complete

```

As you can see, etherweaver operated idempotently, only applying the changes from the desired state that didn't match the current state. This allows you to easily manage and monitor config drift from within your environment.

2.4.2 CLAG

The cumulus switches we have been using as an example also support a feature known as Clustering Link Aggregation, or CLAG.

This allows two independent switches to share link aggregation groups without a single point of failure, and without stacking. This is an excellent use case for fabric inheritance, as there are attributes that the switches share, as well as plenty that they don't. Here is an example CLAG configuration with etherweaver:

```

fabrics:
  network:
    vlans:
      10:
        Name: Employee
      11:
        Name: Servers
      12:
        Name: VOIP
      20:
        Name: Public Wireless
      254:
        Name: Management
    port_profiles:
      OfficePorts:
        untagged_vlan: 10
        tagged_vlans: [12]
      WirelessAccessPoint:
        untagged_vlan: 254

```

(continues on next page)

(continued from previous page)

```

    tagged_vlans: [10, 20]
    Trunk:
      tagged_vlans: [10, 12, 20, 254]
spine_cluster:
  fabric: network
  clag:
    shared_mac: 44:38:39:FF:01:01
  interfaces:
    1G:
      9-10:
        bond_slave: peerlink
      1:
        bond_slave: po1
      2:
        bond_slave: po2
    bond:
      po1:
        clag_id: 1
        profile: Trunk
      po2:
        clag_id: 2
        profile: Trunk
appliances:
  spine1:
    fabric: spine_cluster
    hostname: dist1
    plugin_package: cumulus
    connections:
      ssh:
        hostname: 192.168.122.89
        username: cumulus
        port: 22
      clag:
        priority: 1000
        backup_ip: 192.168.122.18
        clag_cidr: [169.254.2.1/30]
        peer_ip: 169.254.2.2
  spine2:
    fabric: spine_cluster
    hostname: dist2
    plugin_package: cumulus
    connections:
      ssh:
        hostname: 192.168.122.18
        username: cumulus
        port: 22
      clag:
        priority: 0
        backup_ip: 192.168.122.89
        clag_cidr: [169.254.2.2/30]
        peer_ip: 169.254.2.1

```

Applying this state file looks like this:

```

$ etherweaver "*" state.apply --yaml=clag_example.yaml
If you continue, the following changes will be applied:
#####

```

(continues on next page)

(continued from previous page)

```

{  'spine1': [  'net add hostname dist1',
                'net add bridge bridge vids 10-12,20,254',
                'net add bond peerlink bond slaves swp9',
                'net add bond peerlink bond slaves swp10',
                'net add bond po1 bond slaves swp1',
                'net add bond po2 bond slaves swp2',
                'net add interface peerlink.4094 clag backup-ip '
                '192.168.122.18',
                'net add interface peerlink.4094 ip address 169.254.2.1/30',
                'net add interface peerlink.4094 clag peer-ip 169.254.2.2',
                'net add interface peerlink.4094 clag priority 1000',
                'net add interface peerlink.4094 clag sys-mac '
                '44:38:39:FF:01:01',
                'net add bond po1 bridge vids 10,12,20,254',
                'net add bond po2 bridge vids 10,12,20,254'],
    'spine2': [  'net add hostname dist2',
                'net add bridge bridge vids 10-12,20,254',
                'net add bond peerlink bond slaves swp9',
                'net add bond peerlink bond slaves swp10',
                'net add bond po1 bond slaves swp1',
                'net add bond po2 bond slaves swp2',
                'net add interface peerlink.4094 clag backup-ip '
                '192.168.122.89',
                'net add interface peerlink.4094 ip address 169.254.2.2/30',
                'net add interface peerlink.4094 clag peer-ip 169.254.2.1',
                'net add interface peerlink.4094 clag priority 0',
                'net add interface peerlink.4094 clag sys-mac '
                '44:38:39:FF:01:01',
                'net add bond po1 bridge vids 10,12,20,254',
                'net add bond po2 bridge vids 10,12,20,254']]
Do you want to continue? y/[n]y
spine1: 100%|| 13/13 [00:01<00:00, 7.17it/s]
spine2: 100%|| 13/13 [00:06<00:00, 5.02it/s]

```

Run complete

Note: When testing state files in a virtual environment such as GNS3 or vagrant, the cumulus switches clag will not function unless you add 'clagd-args -vm' to /etc/network/interfaces under the peerlink.4094 interface section on both switches.

Configuration Elements

3.1 Nodes

Note: This is a stand-in for some custom auto-documentation I haven't dealt with yet. If you are interested in tackling this, take a look at this [Issue](#).

3.1.1 Main Nodes

```
{
    'dstate': {
        'apply': self.push_state,
        'get': self.dstate,
        'allowed_functions': ['get', 'apply'],
        'description': "Interact with the desired state of the_
↪appliance, either viewing the desired state"
        " or applying it non-interactively."
    },
    'cstate': {
        'get': self.cstate,
        'allowed_functions': ['get']
    },
    'hostname': {
        'set': self.plugin.set_hostname,
        'get': self.plugin.cstate['hostname'],
        'data_type': str,
        'allowed_functions': ['get', 'set', 'del']
    },
    'vlans': {
        'get': self.plugin.cstate['vlans'],
        'set': self.plugin.set_vlans,
```

(continues on next page)

(continued from previous page)

```

        'allowed_functions': ['get', 'set', 'add', 'del']
    },
    'clag': {
        'get': self.cstate['clag'],
        'allowed_functions': ['get'],
        'shared_mac': {
            'get': self.cstate['clag']['shared_mac'],
            'set': self.plugin.set_clag_shared_mac,
            'data_type': str,
            'allowed_functions': ['get', 'set', 'del']
        },
        'priority': {
            'get': self.cstate['clag']['priority'],
            'set': self.plugin.set_clag_priority,
            'data_type': int,
            'allowed_functions': ['get', 'set', 'del']
        },
        'backup_ip': {
            'get': self.cstate['clag']['backup_ip'],
            'set': self.plugin.set_clag_backup_ip,
            'data_type': str,
            'allowed_functions': ['get', 'set', 'del']
        },
        'clag_cidr': {
            'get': self.cstate['clag']['clag_cidr'],
            'set': self.plugin.set_clag_cidr,
            'data_type': list,
            'list_subtype': str,
            'allowed_functions': ['get', 'set', 'del', 'add']
        },
        'peer_ip': {
            'get': self.cstate['clag']['peer_ip'],
            'set': self.plugin.set_clag_peer_ip,
            'data_type': str,
            'allowed_functions': ['get', 'set', 'del']
        }
    },
    'interfaces': {
        'get': self.cstate['interfaces'],
        'allowed_functions': ['get'],
        '1G': {
            'get': self.cstate['interfaces']['1G'],
            'allowed_functions': ['get']
        },
        '10G': {
            'get': self.cstate['interfaces']['10G'],
            'allowed_functions': ['get']
        },
        '40G': {
            'get': self.cstate['interfaces']['40G'],
            'allowed_functions': ['get']
        },
        '100G': {
            'get': self.cstate['interfaces']['100G'],
            'allowed_functions': ['get']
        },
        'bond': {

```

(continues on next page)

(continued from previous page)

```

        'get': self.cstate['interfaces']['bond'],
        'allowed_functions': ['get']
    },
    },
    'protocols': {
        'ntp': {
            'get': self.cstate['protocols']['ntp'],
            'allowed_functions': ['get'],
            'client': {
                {
                    'timezone': {
                        'get': self.plugin.cstate[
↪ 'protocols']['ntp']['client']['timezone'],
                        'set': self.plugin.set_
↪ ntp_client_timezone,
                        'data_type': str,
                        'allowed_functions': ['get
↪ ', 'set']
                    },
                    'servers': {
                        'get': self.plugin.cstate[
↪ 'protocols']['ntp']['client']['servers'],
                        'set': self.plugin.set_
↪ ntp_client_servers,
                        'data_type': list,
                        'list_subtype': str,
                        'allowed_functions': ['get
↪ ', 'set', 'del', 'add']
                    },
                    'get': self.plugin.cstate[
↪ 'protocols']['ntp']['client'],
                    'allowed_functions': ['get']
                }
            },
            'dns': {
                'get': self.plugin.cstate['protocols']['dns'],
                'allowed_functions': ['get'],
                'nameservers': {
                    'get': self.plugin.cstate['protocols']['
↪ 'dns']['nameservers'],
                    'set': self.plugin.set_dns_nameservers,
                    'data_type': list,
                    'data_subtype': str,
                    'allowed_functions': ['get', 'set', 'del',
↪ 'add']
                }
            }
        }
    }
}

```

3.1.2 Interface Nodes

```

{
    'get': int_cstate,
    'set': self.plugin.set_interface,

```

(continues on next page)

(continued from previous page)

```

        'allowed_functions': ['set', 'get'],
        'ip': {
            'get': int_cstate['ip'],
            'allowed_functions': ['get'],
            'addresses': {
                'get': int_cstate['ip']['addresses'],
                'set': self.plugin.set_interface_ip_addresses,
                'data_type': list,
                'list_subtype': str,
                'allowed_functions': ['get', 'add', 'set',
→ 'del']
            }
        },
        'untagged_vlan': {
            'get': int_cstate['untagged_vlan'],
            'set': self.plugin.set_interface_untagged_vlan,
            'data_type': int,
            'allowed_functions': ['get', 'set', 'del']
        },
        'tagged_vlans': {
            'get': int_cstate['tagged_vlans'],
            'set': self.plugin.set_interface_tagged_vlans,
            'data_type': list,
            'list_subtype': int,
            'allowed_functions': ['get', 'set', 'add', 'del']
        }
    }
}

```

Interface Specific Nodes

```

{
    'stp': {
        'get': int_cstate['stp'],
        'allowed_functions': ['get'],
        'port_fast': {
            'get': int_cstate['stp']['port_fast'],
            'set': self.plugin.set_portfast,
            'allowed_functions': ['get', 'set']
        }
    },
    'bond_slave': {
        'get': int_cstate['bond_slave'],
        'set': self.plugin.set_bond_slaves,
        'data_type': str,
        'allowed_functions': ['get', 'delete', 'set']
    },
    'mtu': {
        'get': int_cstate['mtu'],
        'set': self.plugin.set_interface_mtu,
        'data_type': str,
        'allowed_functions': ['get', 'set', 'del']
    }
}

```

Bond Specific Nodes

```
{
    'clag_id': {
        'get': int_cstate['clag_id'],
        'set': self.plugin.set_bond_clag_id,
        'data_type': int,
        'allowed_functions': ['get', 'set', 'del']
    },
    'mtu': {
        'get': int_cstate['mtu'],
        'set': self.plugin.set_bond_mtu,
        'data_type': str,
        'allowed_functions': ['get', 'set', 'del']
    }
}
```

3.2 List Expansion

Wherever lists are valid values, items following the pattern ‘1-3’ will be expanded.

For instance: [1, 3, 7-9] will be expanded to [1, 3, 7, 8, 9]

3.3 Interface Profiles

If any object in the inheritance chain contains a port_profiles node, this profile can be referenced either at the same level or in a child object. This allows you to re-use common port configurations. You can stop the application of profiles on a lower level by defining ‘profile: false’.

For example:

```
fabrics:
  network1:
    vlans:
      4-10:
        fabric: toplevelnet
    port_profiles:
      access:
        untagged_vlan: 1
      wap_trunk:
        untagged_vlan: 2
        tagged_vlans: [4-6, 10]

roles:
  dist1:
    fabric: network1
  interfaces:
    1G:
      4-5:
        profile: access
    1:
      profile: wap_trunk
```

(continues on next page)

(continued from previous page)

```
appliances:
  sw1:
    interfaces:
      5:
        profile: false
        untagged_vlan: 10
    role: dist1
    plugin_package: cumulus
    connections:
      ssh:
        hostname: 10.5.5.33
        username: cumulus
        password: CumulusLinux!
        port: 22
```

3.4 Iterator Keys

Iterator keys are used for dynamic naming of parts of a string value, or an int value. Iterator keys lets you significantly reduce the number of lines needed to implement certain functionality homogeneously across many switchports. Right now, the only iterator key is ‘\$i’, and it represents either the number ID, or number component of the port or bond respectively.

For example:

```
interfaces:
  1G:
    1-5:
      bond_slave: po$i
  bond:
    po1-5:
      clag_id: $i
```

Is equivalent to:

```
interfaces:
  1G:
    1:
      bond_slave: po1
    2:
      bond_slave: po2
    3:
      bond_slave: po3
    4:
      bond_slave: po4
    5:
      bond_slave: po5
  bond:
    po1:
      clag_id: 1
    po2:
      clag_id: 2
    po3:
      clag_id: 3
    po4:
```

(continues on next page)

(continued from previous page)

```
    clag_id: 4
po5:
    clag_id: 5
```


This is the best place to start if you plan on building a plugin.

There are a few different archetypes of each set function:

#TODO Examine archetypes

All methods must have `execute` and it must default to `True`. If `execute` is `True`, run and apply the command immediately. Otherwise just return the line(s) needed to run the command.

If you plan on implementing a method in the future, please override that module and raise a `NotImplemented` exception.

The plugin you implement *MUST* accept all of the arguments present in the skeleton class (documented below).

4.1 Plugin Class Structure

```
class etherweaver.plugins.plugin_class.NetWeaverPlugin
```

```
    add_command (commands)
```

```
        Adds a command to the command queue :param commands: :return:
```

```
    after_connect ()
```

```
        PLUGIN_OVERRIDE
```

```
        Put anything here that your plugin needs to do after a self.connect is called
```

```
    build_ssh_session ()
```

```
        Build the SSH Object
```

```
    connect ()
```

```
        Examine protocol attribute and set up connection accordingly
```

```
    set_bond_admin_down (int_type, bond, down_status, commit=True, execute=True)
```

Parameters

- **int_type** – This is the type of the interface, for instance: ‘bond’, ‘1G’, ‘10G’. Used to determine the group of the interface to be modified.
- **bond** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **down_status** – True for a downed bond, False for an up bond.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_bond_clag_id (*int_type, interface, clag_id, execute=True, delete=False, commit=True*)

Sets the CLAG ID of a bond

Parameters

- **int_type** – This is the type of the interface, for instance: ‘bond’, ‘1G’, ‘10G’. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **clag_id** – Non negative integer
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_bond_mtu (*int_type, bond, mtu, execute=True, commit=True, delete=False*)

Parameters

- **int_type** – The type/speed of the bond (Will always be a string ‘bond’, this parameter exists for consistency)
- **bond** – The ID of the bond
- **mtu** – Non zero non negative integer
- **execute** – If execute is True, this method must run and apply the configuration
- **commit** – If commit is true, this method must also commit the change (if applicable)
- **delete** – Resets MTU to default on appliances that support it. Raise an error if yours doesn’t.

Returns List of commands that can be run to effect the change. You must return the list even if execute=True

set_bond_slaves (*int_type, interface, bond, execute=True, commit=True, delete=False*)

Parameters

- **int_type** – This is the type of the interface, for instance: ‘bond’, ‘1G’, ‘10G’. Used to determine the group of the interface to be modified.

- **interface** – This is the number of the interface. You will likely need to translate this.
- **bond** – This is the ID of the bond that the interface will be added to.
- **execute** – If execute is True, this method must run and apply the configuration
- **commit** – If commit is true, this method must also commit the change (if applicable)
- **delete** – Delete the bond.

Returns List of commands that can be run to effect the change. You must return the list even if execute=True

set_clag_backup_ip (*backup_ip, execute=True, delete=False, commit=True*)
Sets the backup peer IP for CLAG

Parameters

- **backup_ip** – IPv4 or IPv6 backup address of CLAG member (call self._not_supported(string) with string as a helpful error message if your plugin's appliance doesn't support IPv6.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_clag_cidr (*cidr, execute=True, delete=False, commit=True, add=False*)
Set the IP address and subnet mask of the primary CLAG peer interface

Parameters

- **cidr** – CIDR of the appliance's CLAG peering interface as a string. EX; '169.254.2.1/30'
- **add** – List of cidrs to add.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_clag_peer_ip (*peer_ip, execute=True, delete=False, commit=True*)
Sets CLAG interface peer IP

Parameters

- **peer_ip** – IP address of the peer as a string I.E. '169.254.2.1'
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.

- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_clag_priority (*priority, execute=True, delete=False, commit=True*)

Sets the CLAG priority of the appliance

Parameters

- **priority** – Non negative integer
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_clag_shared_mac (*shared_mac, execute=True, delete=False, commit=True*)

Sets the shared mac of the CLAG daemon on the appliance

Parameters

- **shared_mac** – String containing the shared MAC address of the cluster
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_dns_nameservers (*nameserverlist, execute=True, commit=True, delete=False, add=False*)

This method sets the dns resolvers used by the appliance.

Parameters

- **nameserverlist** – A list of DNS server IPv4 or IPv6 addresses to add to the resolver list.
- **execute** – If execute is True, this method must run and apply the configuration.
- **add** – List of DNS nameservers to be added.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all tagged vlans from the interface. If delete is true and there is one or more values, remove only the specified values.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_hostname (*hostname, execute=True, commit=True, delete=False*)

This method sets the system hostname of the appliance

Parameters

- **hostname** – A string containing the hostname of the system. Either a FQDN or short name, but not both (as most appliances don't differentiate)
- **execute** – If execute is True, this method must run and apply the configuration.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all tagged vlans from the interface. If delete is true and there is one or more values, remove only the specified values.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_interface_admin_down (*int_type, interface, down_status, commit=True, execute=True*)

Parameters

- **int_type** – This is the type of the interface, for instance: 'bond', '1G', '10G'. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **down_status** – True for a downed interface, False for an up interface.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_interface_ip_addresses (*int_type, interface, ips, execute=True, commit=True, delete=False, add=False*)

Adds and removes IP addresses from the interface

Parameters

- **int_type** – This is the type of the interface, for instance: 'bond', '1G', '10G'. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **ips** – A list of CIDR notated IP addresses (I.E. ['10.0.0.1/16', '192.168.0.5/24'])
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_interface_mtu (*int_type, interface, mtu, execute=True, commit=True, delete=False*)

Parameters

- **int_type** – The type/speed of interface
- **interface** – The ID of the interface

- **mtu** – Non zero non negative integer
- **execute** – If execute is True, this method must run and apply the configuration
- **commit** – If commit is true, this method must also commit the change (if applicable)
- **delete** – Resets MTU to default on appliances that support it. Raise an error if yours doesn't.

Returns List of commands that can be run to effect the change. You must return the list even if `execute=True`

set_interface_tagged_vlans (*speed, interface, vlans, execute=True, commit=True, delete=False, add=False*)

This method modifies the list of allowed tagged vlans for a given interface.

Parameters

- **speed** – This is the type of the interface, for instance: 'bond', '1G', '10G'. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **vlans** – This parameter will always contain a list of vlans to add or remove, even if there is a single value, and may be empty.
- **execute** – If execute is True, this method must run and apply the configuration.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all tagged vlans from the interface. If delete is true and there is one or more values, remove only the specified values.
- **add** – If add is true and value is set, add all the tagged vlans in the list without removing any.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF `execute=True`

set_interface_untagged_vlan (*int_type, interface, vlan, execute=True, delete=False, commit=True*)

Sets the untagged (PVID) of an interface

Parameters

- **int_type** – This is the type of the interface, for instance: 'bond', '1G', '10G'. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **vlan** – The ID of the vlan to add.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.
- **execute** – If execute is True, this method must run and apply the configuration.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF `execute=True`

set_ntp_client_servers (*ntpserverlist, execute=True, commit=True, delete=False*)

Parameters

- **ntpserverlist** – A list of ntp servers to add (IP Addresses or DNS)
- **execute** – If execute is True, this method must run and apply the configuration.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all values. If delete is true and there is one or more values, remove only the specified values.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_ntp_client_timezone (*timezone, execute=True, delete=True, commit=True*)

Sets the timezone of the NTP client.

Parameters

- **timezone** – Contains a string containing a valid timeozone recognized by pytz
- **execute** – If execute is True, this method must run and apply the configuration.
- **commit** – If commit is true, the appliance must load the new configuration as part of this method.
- **delete** – If delete is true and no value is set, remove all tagged vlans from the interface. If delete is true and there is one or more values, remove only the specified values. If you cannot delete the timezone on your appliance (I.E., one must always be specified), call the inherited `self._not_supported(string)`, with string being an informative error message.

Returns Return the list of commands that can be run to effect the change. You must return the list EVEN IF execute=True

set_portfast (*int_type, interface, enable_bool, execute=True, commit=True*)

Portfast skips STP listening stage on the port to allow the link to come up faster.

Parameters

- **int_type** – This is the type of the interface, for instance: 'bond', '1G', '10G'. Used to determine the group of the interface to be modified.
- **interface** – This is the number of the interface, or text ID of the bond. You will likely need to translate this.
- **enable_bool** – If true, enable portfast, if false disable portfast
- **execute** – If execute is True, this method must run and apply the configuration
- **commit** – If commit is true, this method must also commit the change (if applicable)

Returns List of commands that can be run to effect the change. You must return the list even if execute=True

CHAPTER 5

License

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether

(continues on next page)

(continued from previous page)

gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

(continues on next page)

(continued from previous page)

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding

(continues on next page)

(continued from previous page)

Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,

(continues on next page)

(continued from previous page)

and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices".
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the

(continues on next page)

(continued from previous page)

product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as

(continues on next page)

(continued from previous page)

part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or

(continues on next page)

(continued from previous page)

authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

(continues on next page)

(continued from previous page)

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

(continues on next page)

(continued from previous page)

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you

(continues on next page)

(continued from previous page)

to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY

(continues on next page)

(continued from previous page)

GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands

(continues on next page)

(continued from previous page)

might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read [<http://www.gnu.org/philosophy/why-not-lgpl.html>](http://www.gnu.org/philosophy/why-not-lgpl.html).

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

A

`add_command()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 21

`after_connect()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 21

B

`build_ssh_session()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 21

C

`connect()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 21

N

`NetWeaverPlugin` (class in etherweaver.plugins.plugin_class), 21

S

`set_bond_admin_down()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 21

`set_bond_clag_id()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 22

`set_bond_mtu()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 22

`set_bond_slaves()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 22

`set_clag_backup_ip()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 23

`set_clag_cidr()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 23

`set_clag_peer_ip()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 23

`set_clag_priority()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 24

`set_clag_shared_mac()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 24

`set_dns_nameservers()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 24

`set_hostname()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 24

`set_interface_admin_down()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 25

`set_interface_ip_addresses()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 25

`set_interface_mtu()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 25

`set_interface_tagged_vlans()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 26

`set_interface_untagged_vlan()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 26

`set_ntp_client_servers()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 26

`set_ntp_client_timezone()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 27

`set_portfast()` (etherweaver.plugins.plugin_class.NetWeaverPlugin method), 27