

CISCO SYSTEMS - CX CENTERS ENTERPRISE NETWORKING



TOTD: New Configuration Archive feature -
 Export Device configurations to an
 encrypted zip file

Prepared for: Cisco DNA Customer, Solutions Support

Prepared by: Tomas de Leon, Technical Leader

September 21, 2020

Document number: 09212020_v1

CISCO SYSTEMS - CX CENTERS ENTERPRISE NETWORKING

TECHNOTE OF THE DAY (TOTD) -- NEW CONFIGURATION ARCHIVE FEATURE - EXPORT DEVICE CONFIGURATIONS TO AN ENCRYPTED ZIP FILE IN CISCO DNA CENTER VERSION 2.1.2.X

Objective

The objective of this document is to show users the new Configuration Archive feature. A new Rest API was added to export device configurations to an encrypted zip file. The new API was added in Cisco DNA Center version 2.1.2.0.

Goals

Provide an awareness of the new configuration archive feature and to provide an example of using the API with the Postman Rest API Client.

The following technote is written against the Release of Cisco DNA Center version 2.1.2.0.

Reference Information:

- Cisco DNA Center version 2.1.2.x0
- Postman Version 7.32.0 (7.32.0)

CISCO SYSTEMS - CX CENTERS ENTERPRISE NETWORKING

CONFIGURATION ARCHIVE FEATURE

In the Release of Cisco DNA Center version 2.1.2.0, there is only one API URL task provided for the Configuration Archive feature. The current API task is to export device configurations to an encrypted zip file. My understanding is that more API URL task will be coming in future releases but this is always subject to change.

Platform . Developer Toolkit . APIs

Configuration Archive

<u>Name</u>	<u>Description</u>
<i>Export Device configurations</i>	<i>Export Device configurations to an encrypted zip file.</i>
<u>Method</u>	<u>URL</u>
POST	<a href="https://<_dnac.ip.address_>/dna/intent/api/v1/network-device-archive/cleartext">https://<_dnac.ip.address_>/dna/intent/api/v1/network-device-archive/cleartext

Model Schema

payload:

```
{  
  
  "deviceId": "<List of device uuids to be exported>",  
  
  "password": "<Password to export the file. Same should be used to view the file>"  
}
```

Note: The schema payload is not listed in the Cisco DNA Center screens or the help. A CDET has been created to add this documentation to the UI in an upcoming release.

Platform . Developer Toolkit . APIs (cont.)

Cisco DNA Center UI Example:

The screenshot shows the Cisco DNA Center Platform Developer Toolkit interface. The 'APIs' tab is selected, and the 'Site Management' category is expanded. Under 'Configuration Archive', a table lists the API details:

Method	Name	Description	URL
POST	Export Device configurations	Export Device configurations to an encrypted zip file.	/network-device-archive/cleartext

The screenshot shows a modal window titled 'Export Device configurations'. It provides detailed information about the API:

- DESCRIPTION:** Export Device configurations to an encrypted zip file.
- Method:** POST
- URL:** https://172.18.217.184/dna/intent/api/v1/network-device-archive/cleartext
- PARAMETERS:** Request Headers Parameters

Name	Description	DataType	Default Value	Required
Content-Type		string	application/json	true
- RESPONSES:** Response Codes

Code	Message
200	

Try It

Method	Public URL :
POST	https://172.18.217.184/dna/intent/api/v1/network-device-archive/cleartext

Name	Description	Value
Headers		
Content-Type*		application/json

Body

```

1 {
2
3   "deviceId": "<List of device uuids to be exported>",
4
5   "password": "<Password to export the file. Same should be used to view the file>"
6
7 }
```

Cancel
Run

Note: The schema payload is not listed in the Cisco DNA Center screens or the help. A CDET has been created to add this documentation to the UI in an upcoming release.

USING THE POSTMAN REST API CLIENT FOR EXPORTING DEVICE CONFIGURATIONS

Use Case Scenario - Export Device configurations to an encrypted zip file using the Postman Rest API Client.

- Cisco DNA Center version 2.1.2.0
- Postman Version 7.32.0 (7.32.0)

STEP 1 - GET TOKEN - Authentication/Authorization

<u>Method</u>	<u>URL</u>
POST	https://<_dnac.ip.address_>/dna/system/api/v1/auth/token

Response:

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI1ZjU3OWQwNWQxZDZkNTAwOGJlOTdhNWYiLCJhdXRoU291cmNLIjoizXh0ZXJuYmwiLCJ0ZW5hbnR0YXV1IjoieVE5UMCI6IjVjbGVzIjpbIjVlNDc0ODJhN2E4MDQ3MDBiZjhiZTB1ZiJdLCJ0ZW5hbnRJCi6IjVlNDc0ODJhN2E4MDQ3MDBiZjhiZTB1ZCI6ImV4cCI6MTYwMDYxNTI2MywiaWF0IjoxNjAwNjExNjYzLCJqdGkiOiJjYjA0NDQxMi1jYjRiLTQ4ZjQ0TE2YS1iMGY4YmQzZmM5NmIiLCJ1c2VybmFtZSI6InYxYWRTaW4ifQ.hEKDnflrGfx10-H9H0HLJIGSs000BgpCp_C8a7R8LIk_gjMmuXWxQxomLQmwsG6j3v9xe6Xje9Sh3-_pPv0RjQxteUhsZunGFhhCxbPgSlhpLLhedAjeXAofif0kD1jgGgioRRufbtQMQR4VdoMAHSGh8xHH7RgCPn8r19eFZmS327tIkLRQFfb_yIURp3awpWtCsStPCyztmNxogkjtQALImzJstoKELW_W-FvqDGpHDldQ86x6Z0Xt_Z_IAwAm7reLXe40rxZeP3DGqnxceD_Hr50Vzq2xY_lWd2knH71ZGjLIYMKiNBAMpppFbfpsvVLAxgy4WzDIkq5ibg
```

The screenshot shows the Postman interface for a POST request to `https://172.18.217.184/dna/system/api/v1/auth/token`. The request is configured with Basic Auth, Username `v1admin`, and Password `.....`. The response status is `200 OK` with a response time of `900 ms` and a size of `1.18 KB`. The response body is a JSON object containing a token:

```
{
  "Token": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI1ZjU3OWQwNWQxZDZkNTAwOGJlOTdhNWYiLCJhdXRoU291cmNLIjoizXh0ZXJuYmwiLCJ0ZW5hbnR0YXV1IjoieVE5UMCI6IjVjbGVzIjpbIjVlNDc0ODJhN2E4MDQ3MDBiZjhiZTB1ZiJdLCJ0ZW5hbnRJCi6IjVlNDc0ODJhN2E4MDQ3MDBiZjhiZTB1ZCI6ImV4cCI6MTYwMDYxNTI2MywiaWF0IjoxNjAwNjExNjYzLCJqdGkiOiJjYjA0NDQxMi1jYjRiLTQ4ZjQ0TE2YS1iMGY4YmQzZmM5NmIiLCJ1c2VybmFtZSI6InYxYWRTaW4ifQ.hEKDnflrGfx10-H9H0HLJIGSs000BgpCp_C8a7R8LIk_gjMmuXWxQxomLQmwsG6j3v9xe6Xje9Sh3-_pPv0RjQxteUhsZunGFhhCxbPgSlhpLLhedAjeXAofif0kD1jgGgioRRufbtQMQR4VdoMAHSGh8xHH7RgCPn8r19eFZmS327tIkLRQFfb_yIURp3awpWtCsStPCyztmNxogkjtQALImzJstoKELW_W-FvqDGpHDldQ86x6Z0Xt_Z_IAwAm7reLXe40rxZeP3DGqnxceD_Hr50Vzq2xY_lWd2knH71ZGjLIYMKiNBAMpppFbfpsvVLAxgy4WzDIkq5ibg"
}
```

STEP 2 - GET DEVICE INFO using managementIpAddressMethod

GET

URLhttps://<_dnac.ip.address_>/dna/intent/api/v1/network-device?managementIpAddress=<_device.mgmt.ip.address_>

- Test Case Parameters

NETWORK DEVICE INFORMATION

NAME	MGMT IP	PLATFORM	SERIAL_NUM
deadbeef-1.dna.local	192.168.0.101	C9200L-48T-4G	JAE22490HPJ
deadbeef-2.dna.local	192.168.0.102	C9200L-48T-4G	JAE22490HGT

Response: (abbreviated)

```
{
  "response": [
    {
      "family": "Switches and Hubs",
      "collectionStatus": "Managed",
      "serialNumber": "JAE22490HPJ",
      "macAddress": "08:ec:f5:88:0d:80",
      "hostname": "deadbeef-1.dna.local",
      "managementState": "Managed",
      "managementIpAddress": "192.168.0.101",
      "platformId": "C9200L-48T-4G",
      "reachabilityStatus": "Reachable",
      "series": "Cisco Catalyst 9200 Series Switches",
      "snmpContact": "Tomas de Leon - CX 919.867.5309",
      "snmpLocation": "Cisco Systems, North Carolina",
      "deviceSupportLevel": "Supported",
      "softwareType": "IOS-XE",
      "softwareVersion": "17.3.1",
      "inventoryStatusDetail": "<status><general code=\\\"SUCCESS\\\"/></status>",
      "type": "Cisco Catalyst 9200L Switch Stack",
      "role": "ACCESS",
    }
  ]
}
```

```
    "instanceUuid": "2862a8ba-2e2c-4560-8d21-9e6188d862ed",
    "instanceTenantId": "5e47482a7a804700bf8be0ed",
    "id": "2862a8ba-2e2c-4560-8d21-9e6188d862ed"
  }
],
"version": "1.0"
}
```

Response: (abbreviated)

```
{
  "response": [
    {
      "family": "Switches and Hubs",
      "hostname": "deadbeef-2.dna.local",
      "macAddress": "08:ec:f5:93:fb:00",
      "serialNumber": "JAE22490HGT",
      "collectionStatus": "Managed",
      "managementIpAddress": "192.168.0.102",
      "platformId": "C9200L-48T-4G",
      "reachabilityStatus": "Reachable",
      "series": "Cisco Catalyst 9200 Series Switches",
      "snmpContact": "Tomas de Leon - CX 919.867.5309",
      "snmpLocation": "Cisco Systems, North Carolina",
      "inventoryStatusDetail": "<status><general code=\\\"SUCCESS\\\"/></status>",
      "managementState": "Managed",
      "type": "Cisco Catalyst 9200L Switch Stack",
      "role": "ACCESS",
      "deviceSupportLevel": "Supported",
      "softwareType": "IOS-XE",
      "softwareVersion": "17.3.1",
      "instanceTenantId": "5e47482a7a804700bf8be0ed",
      "instanceUuid": "e0526635-4c7d-4908-b82b-563c80597d65",
      "id": "e0526635-4c7d-4908-b82b-563c80597d65"
    }
  ],
  "version": "1.0"
}
```


TOTD -- Config Archive - GET DEVICE INFO using managementIpAddress

Examples BUILD

GET https://172.18.217.184/dna/intent/api/v1/network-device?managementIpAddress=192.168.0.101 Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> managementIpAddress	192.168.0.101	

Body Cookies Headers (14) Test Results Status: 200 OK Time: 2.14 s Size: 2.2 KB Save Response

Pretty Raw Preview Visualize JSON

```

29     "managementIpAddress": "192.168.0.101",
30     "platformId": "C9200L-48T-4G",
31     "reachabilityFailureReason": "",
32     "reachabilityStatus": "Reachable",
33     "series": "Cisco Catalyst 9200 Series Switches",
34     "snmpContact": "Tomas de Leon - CX 919.867.5309",
35     "snmpLocation": "Cisco Systems, North Carolina",
36     "deviceSupportLevel": "Supported",
37     "softwareType": "IOS-XE",
38     "softwareVersion": "17.3.1",
39     "inventoryStatusDetail": "<status>general code=\"SUCCESS\"/></status>",
40     "lastUpdateTime": 1600597632020,
41     "roleSource": "AUTO",
42     "description": "Cisco IOS Software [Amsterdam], Catalyst L3 Switch Software (CAT9K_LITE_IOSXE), Version 17.3.1, RELEASE SOFTWARE (fc5) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2020 by Cisco Systems, Inc. Compiled Fri 07-Aug-20 17:37 by mc",
43     "location": null,
44     "type": "Cisco Catalyst 9200L Switch Stack",
45     "role": "ACCESS",
46     "instanceUuid": "2862a8ba-2e2c-4560-8d21-9e6188d862ed",
47     "instanceTenantId": "5e47482a7a804700bf8be0ed",
48     "id": "2862a8ba-2e2c-4560-8d21-9e6188d862ed"
49   }
50 }
    
```

TOTD -- Config Archive - GET DEVICE INFO using managementIpAddress

Examples BUILD

GET https://172.18.217.184/dna/intent/api/v1/network-device?managementIpAddress=192.168.0.102 Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> managementIpAddress	192.168.0.102	

Body Cookies Headers (14) Test Results Status: 200 OK Time: 3.33 s Size: 2.2 KB Save Response

Pretty Raw Preview Visualize JSON

```

25     "managementIpAddress": "192.168.0.102",
26     "platformId": "C9200L-48T-4G",
27     "reachabilityFailureReason": "",
28     "reachabilityStatus": "Reachable",
29     "series": "Cisco Catalyst 9200 Series Switches",
30     "snmpContact": "Tomas de Leon - CX 919.867.5309",
31     "snmpLocation": "Cisco Systems, North Carolina",
32     "tagCount": "0",
33     "tunnelUdpPort": null,
34     "uptimeSeconds": 2891785,
35     "waasDeviceMode": null,
36     "collectionInterval": "Global Default",
37     "inventoryStatusDetail": "<status>general code=\"SUCCESS\"/></status>",
38     "managementState": "Managed",
39     "location": null,
40     "type": "Cisco Catalyst 9200L Switch Stack",
41     "role": "ACCESS",
42     "deviceSupportLevel": "Supported",
43     "softwareType": "IOS-XE",
44     "softwareVersion": "17.3.1",
45     "roleSource": "AUTO",
46     "instanceTenantId": "5e47482a7a804700bf8be0ed",
47     "instanceUuid": "e0526635-4c7d-4908-b82b-563c80597d65",
48     "id": "e0526635-4c7d-4908-b82b-563c80597d65"
49   }
50 }
    
```

STEP 3 - EXPORT CONFIGURATION using DEVICE_ID

Method URL
POST https://<_dnac.ip.address_>/dna/intent/api/v1/network-device-archive/cleartext

Body

```
{  
  "deviceId": [  
    "2862a8ba-2e2c-4560-8d21-9e6188d862ed",  
    "e0526635-4c7d-4908-b82b-563c80597d65"  
  ],  
  "password": "Cisco123#"  
}
```

Response:

```
{  
  "response": {  
    "taskId": "fa033bb2-81ab-46df-a39c-4455e9410e36",  
    "url": "/api/v1/task/fa033bb2-81ab-46df-a39c-4455e9410e36"  
  }  
}
```

The screenshot displays a REST client interface for a POST request. The request URL is `https://172.18.217.184/dna/intent/api/v1/network-device-archive/cleartext`. The request body is a JSON object with the following content:

```
1 {  
2   "deviceId": [  
3     "2862a8ba-2e2c-4560-8d21-9e6188d862ed",  
4     "e0526635-4c7d-4908-b82b-563c80597d65"  
5   ],  
6   "password": "Cisco123#"  
7 }
```

The response status is 202 Accepted, with a time of 227 ms and a size of 743 B. The response body is a JSON object:

```
1 {  
2   "response": {  
3     "taskId": "fa033bb2-81ab-46df-a39c-4455e9410e36",  
4     "url": "/api/v1/task/fa033bb2-81ab-46df-a39c-4455e9410e36"  
5   },  
6   "version": "1.0"  
7 }
```

STEP 4 - GET FILE INFO using TASK_ID

<u>Method</u>	<u>URL</u>
GET	<a href="https://<_dnac.ip.address_>/api/v1/task/<_task.id_>">https://<_dnac.ip.address_>/api/v1/task/<_task.id_>

From Step 3:

"taskId": "fa033bb2-81ab-46df-a39c-4455e9410e36"

Response: (abbreviated)

```
{
  "response": {
    "version": 1600613167386,
    "endTime": 1600613169009,
    "startTime": 1600613167384,
    "progress": "Device configuration Successfully exported as password protected ZIP.",
    "additionalStatusURL": "/api/v1/file/4962fff5-33e7-49f5-a7b1-b7c9731600b3",
    "lastUpdate": 1600613167386,
  }
}
```

The screenshot shows a REST client interface with the following details:

- Method: GET
- URL: <https://172.18.217.184/api/v1/task/fa033bb2-81ab-46df-a39c-4455e9410e36>
- Send button
- Save button
- Params, Authorization, Headers (9), Body, Pre-request Script, Tests, Settings
- Body tab selected
- Request body: This request does not have a body
- Status: 200 OK, Time: 61 ms, Size: 1.06 KB
- Save Response button
- Response format: Pretty
- Response content (JSON):

```
1 {
2   "response": {
3     "version": 1600613167386,
4     "endTime": 1600613169009,
5     "startTime": 1600613167384,
6     "progress": "Device configuration Successfully exported as password protected ZIP.",
7     "additionalStatusURL": "/api/v1/file/4962fff5-33e7-49f5-a7b1-b7c9731600b3",
8     "lastUpdate": 1600613167386,
9     "isError": false,
10    "rootId": "fa033bb2-81ab-46df-a39c-4455e9410e36",
11    "serviceType": "NCAR",
12    "username": "NCAR88888",
13    "instanceTenantId": "5e47482a7a804700bf8be0ed",
14    "id": "fa033bb2-81ab-46df-a39c-4455e9410e36"
15  },
16  "version": "1.0"
17 }
```

STEP 5 - GET FILE ZIPPED FILE using additionalStatusURL

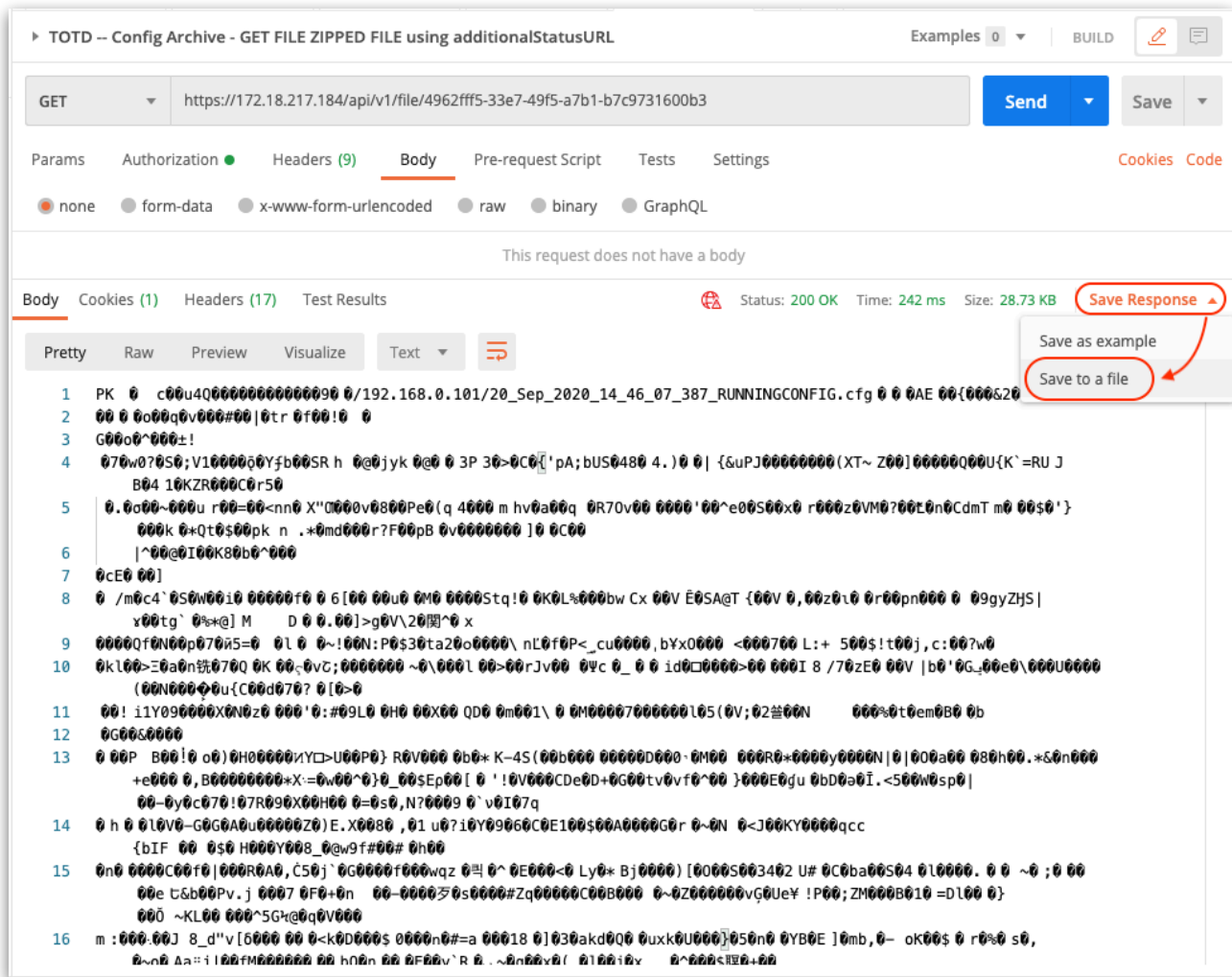
Method URL
GET https://<_dnac.ip.address_>/<_additionalStatusURL.string_>

From Step 4:

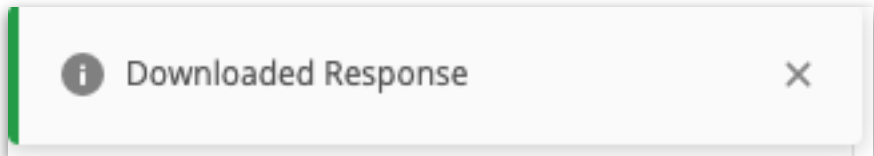
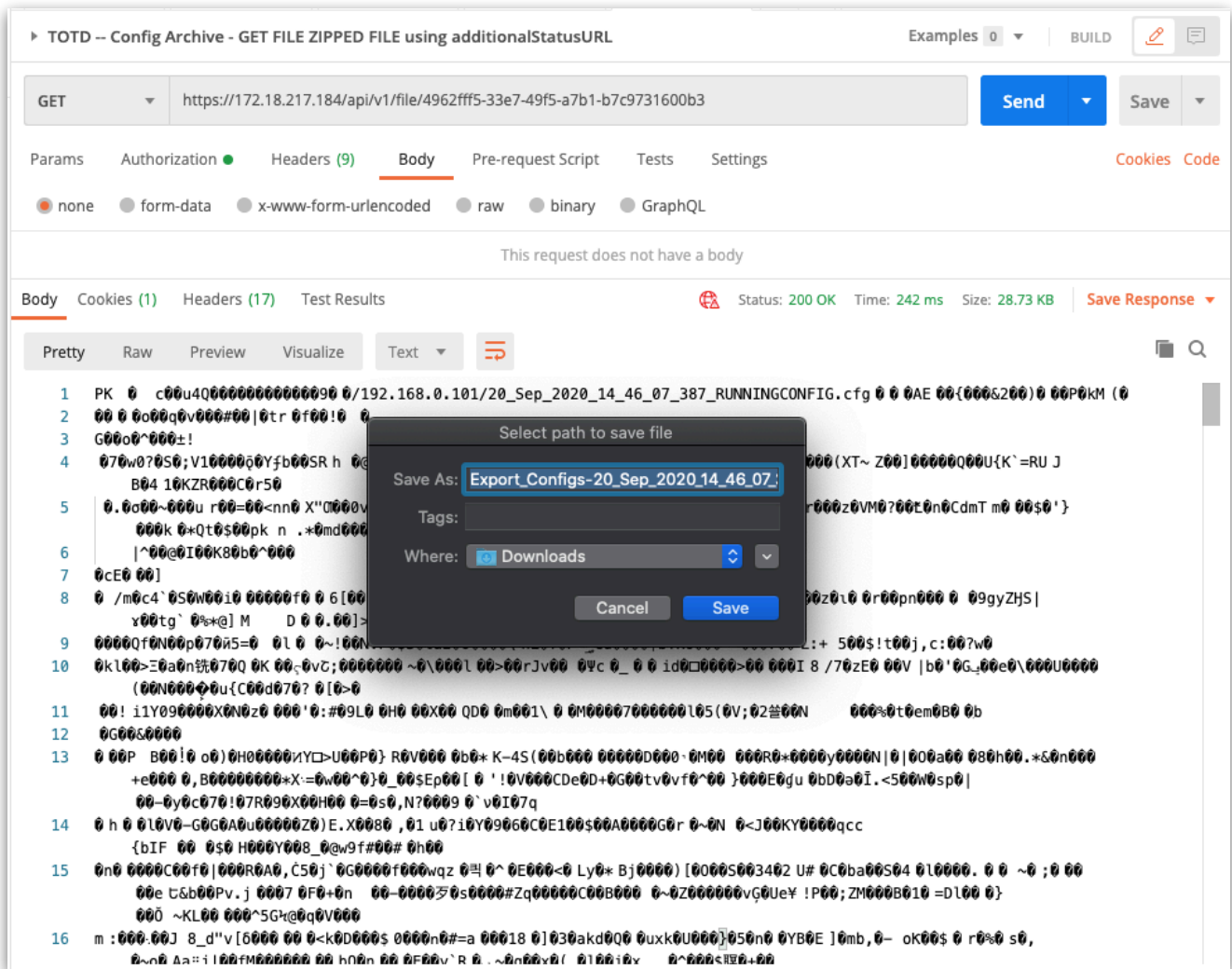
"additionalStatusURL": **"/api/v1/file/4962fff5-33e7-49f5-a7b1-b7c9731600b3"**,

Response: (encoded text & symbols)

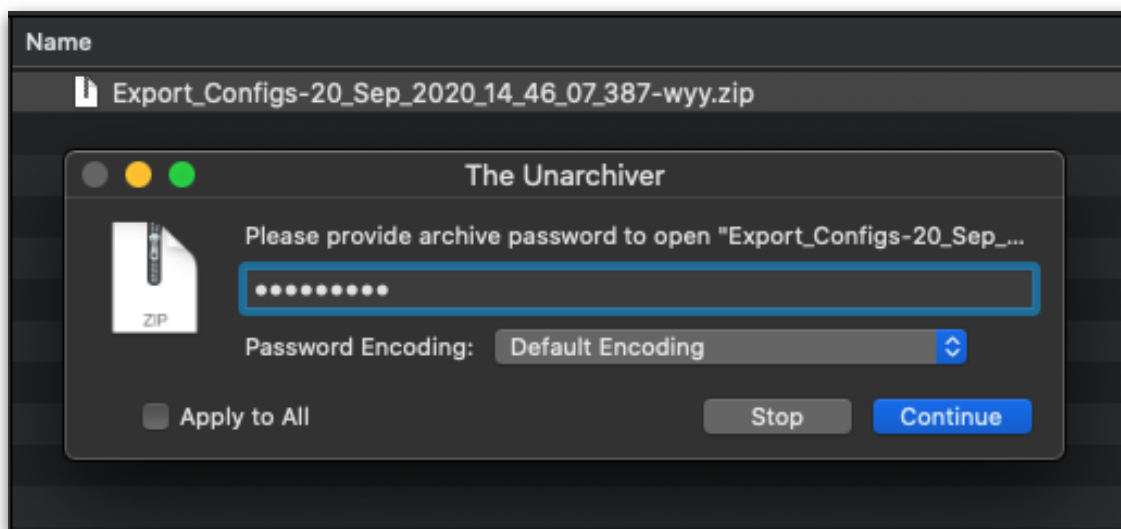
- Click on the "Save Response" Dropdown Arrow and Select "Save To a file"



- Select the [File Name](#), Select the [Download location](#) of the Zipped File, and [Save](#)



- Expand the downloaded zipped file
- Enter the Password used in Step 3



- Enclosed in the Directory for each Device will be the [RUNNING CONFIG](#), [STARTUP CONFIG](#), and [VLAN DATA](#)

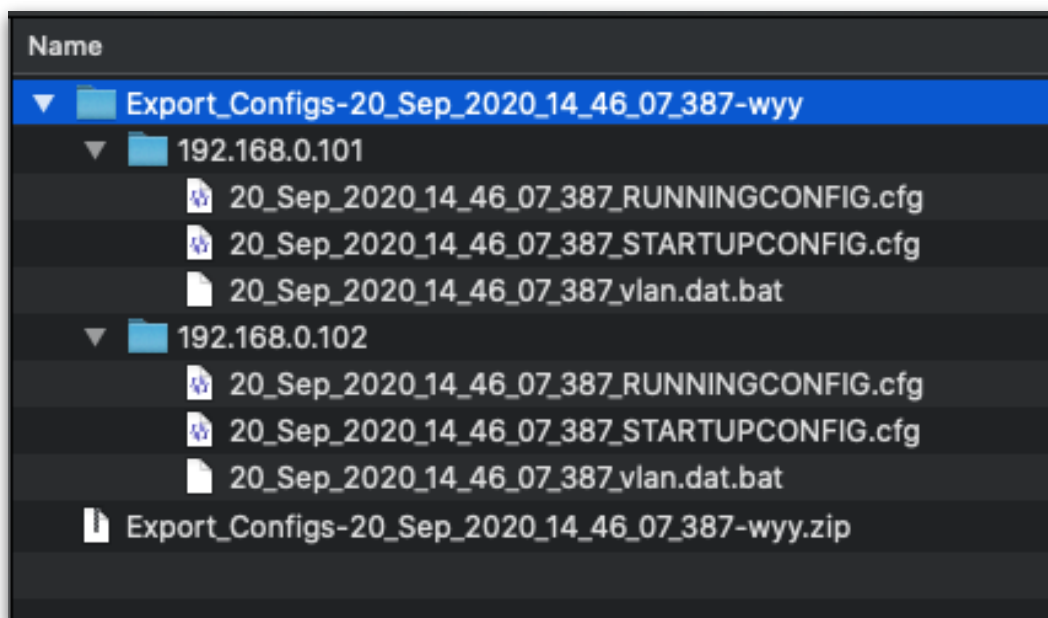
Export_Configs-20_Sep_2020_14_46_07_387-wyy.zip

```
Configuration-Archive $ ls -lR *
-rw-r--r--@ 1 tdeleon  staff  28681 Sep 20 10:52 Export_Configs-20_Sep_2020_14_46_07_387-wyy.zip

Export_Configs-20_Sep_2020_14_46_07_387-wyy:
total 0
drwxr-xr-x@ 5 tdeleon  staff  160 Sep 20 10:58 192.168.0.101
drwxr-xr-x@ 5 tdeleon  staff  160 Sep 20 10:58 192.168.0.102

Export_Configs-20_Sep_2020_14_46_07_387-wyy/192.168.0.101:
total 112
-rwxr-xr-x@ 1 tdeleon  staff  28539 Sep 20 2020 20_Sep_2020_14_46_07_387_RUNNINGCONFIG.cfg
-rwxr-xr-x@ 1 tdeleon  staff  22623 Sep 20 2020 20_Sep_2020_14_46_07_387_STARTUPCONFIG.cfg
-rwxr-xr-x@ 1 tdeleon  staff   556 Sep 20 2020 20_Sep_2020_14_46_07_387_vlan.dat.bat

Export_Configs-20_Sep_2020_14_46_07_387-wyy/192.168.0.102:
total 112
-rwxr-xr-x@ 1 tdeleon  staff  28504 Sep 20 2020 20_Sep_2020_14_46_07_387_RUNNINGCONFIG.cfg
-rwxr-xr-x@ 1 tdeleon  staff  22588 Sep 20 2020 20_Sep_2020_14_46_07_387_STARTUPCONFIG.cfg
-rwxr-xr-x@ 1 tdeleon  staff   804 Sep 20 2020 20_Sep_2020_14_46_07_387_vlan.dat.bat
```



```
Configuration-Archive $ ls -lR *
```

```
-rw-r--r--@ 1 tdeleon  staff  28681 Sep 20 10:52 Export_Configs-20_Sep_2020_14_46_07_387-wyy.zip
```

```
Export_Configs-20_Sep_2020_14_46_07_387-wyy:
```

```
total 0
```

```
drwxr-xr-x@ 5 tdeleon  staff   160 Sep 20 10:58 192.168.0.101
```

```
drwxr-xr-x@ 5 tdeleon  staff   160 Sep 20 10:58 192.168.0.102
```

```
Export_Configs-20_Sep_2020_14_46_07_387-wyy/192.168.0.101:
```

```
total 112
```

```
-rw-r-xr-x@ 1 tdeleon  staff  28539 Sep 20 2020 20_Sep_2020_14_46_07_387_RUNNINGCONFIG.cfg
```

```
-rw-r-xr-x@ 1 tdeleon  staff  22623 Sep 20 2020 20_Sep_2020_14_46_07_387_STARTUPCONFIG.cfg
```

```
-rw-r-xr-x@ 1 tdeleon  staff    556 Sep 20 2020 20_Sep_2020_14_46_07_387_vlan.dat.bat
```

```
Export_Configs-20_Sep_2020_14_46_07_387-wyy/192.168.0.102:
```

```
total 112
```

```
-rw-r-xr-x@ 1 tdeleon  staff  28504 Sep 20 2020 20_Sep_2020_14_46_07_387_RUNNINGCONFIG.cfg
```

```
-rw-r-xr-x@ 1 tdeleon  staff  22588 Sep 20 2020 20_Sep_2020_14_46_07_387_STARTUPCONFIG.cfg
```

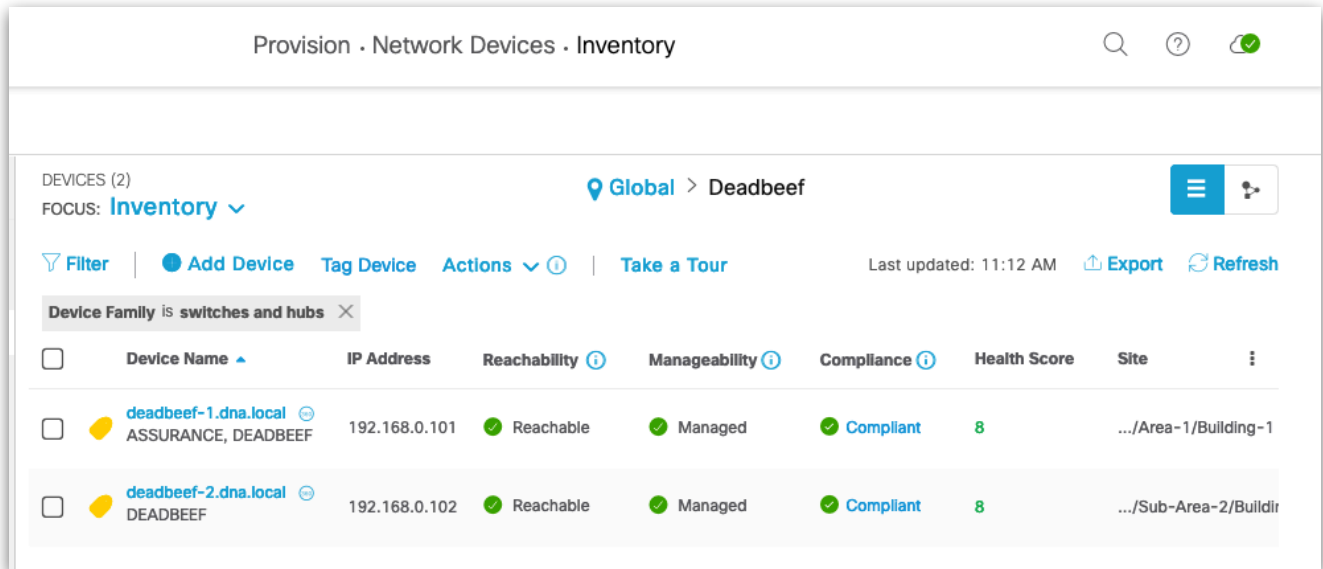
```
-rw-r-xr-x@ 1 tdeleon  staff    804 Sep 20 2020 20_Sep_2020_14_46_07_387_vlan.dat.bat
```

```
Configuration-Archive $ █
```

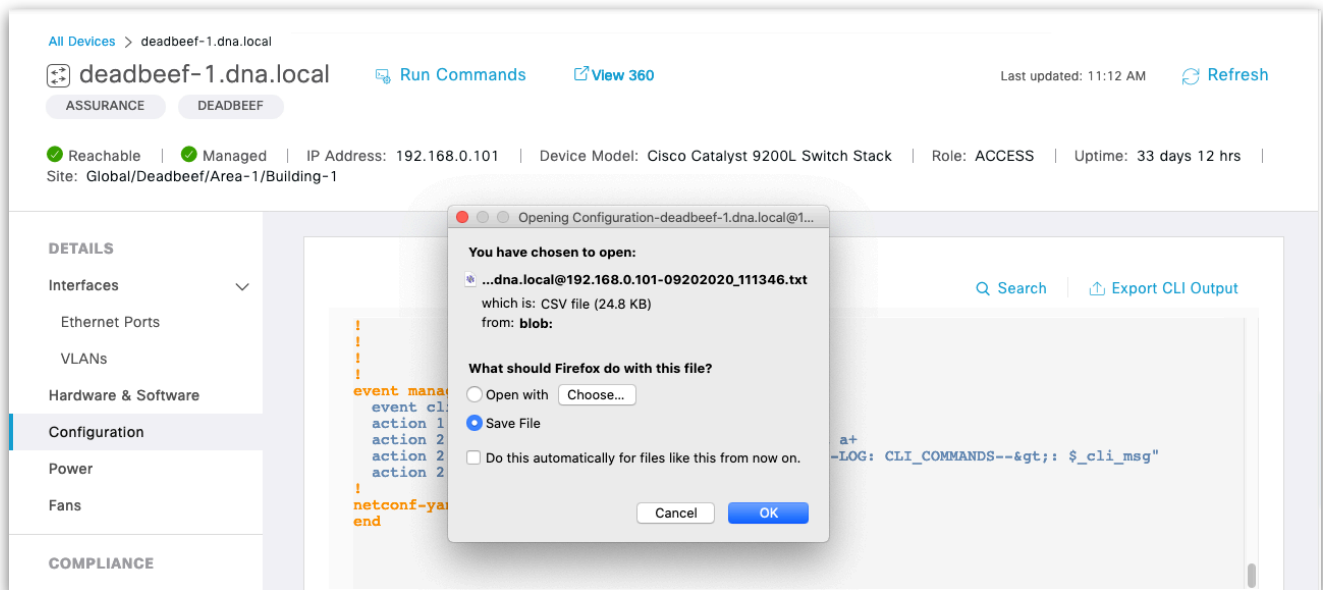
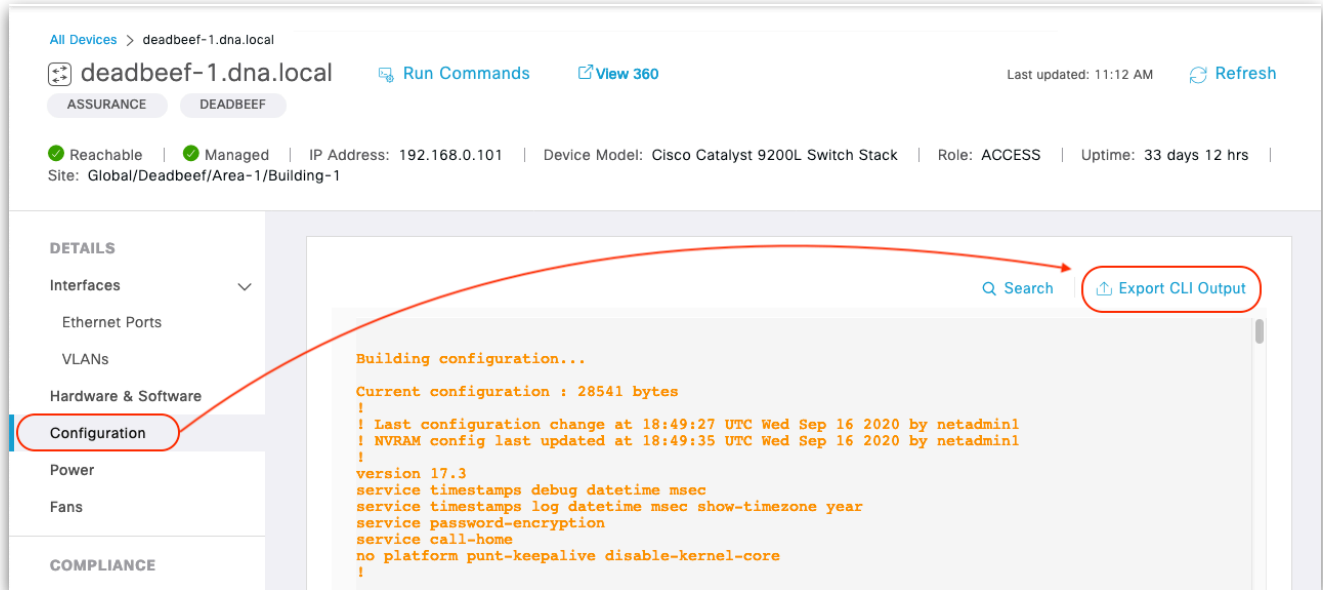
OTHER WAYS TO GET DEVICE CONFIGURATION INFO FROM THE CISCO DNA CENTER UI

Cisco DNA Center UI provides some other ways to get device configuration information:

- Provision.Inventory.Device - **Export CLI Output**
- Provision.Inventory.Device - **Command Runner**



Export CLI Output



Command Runner

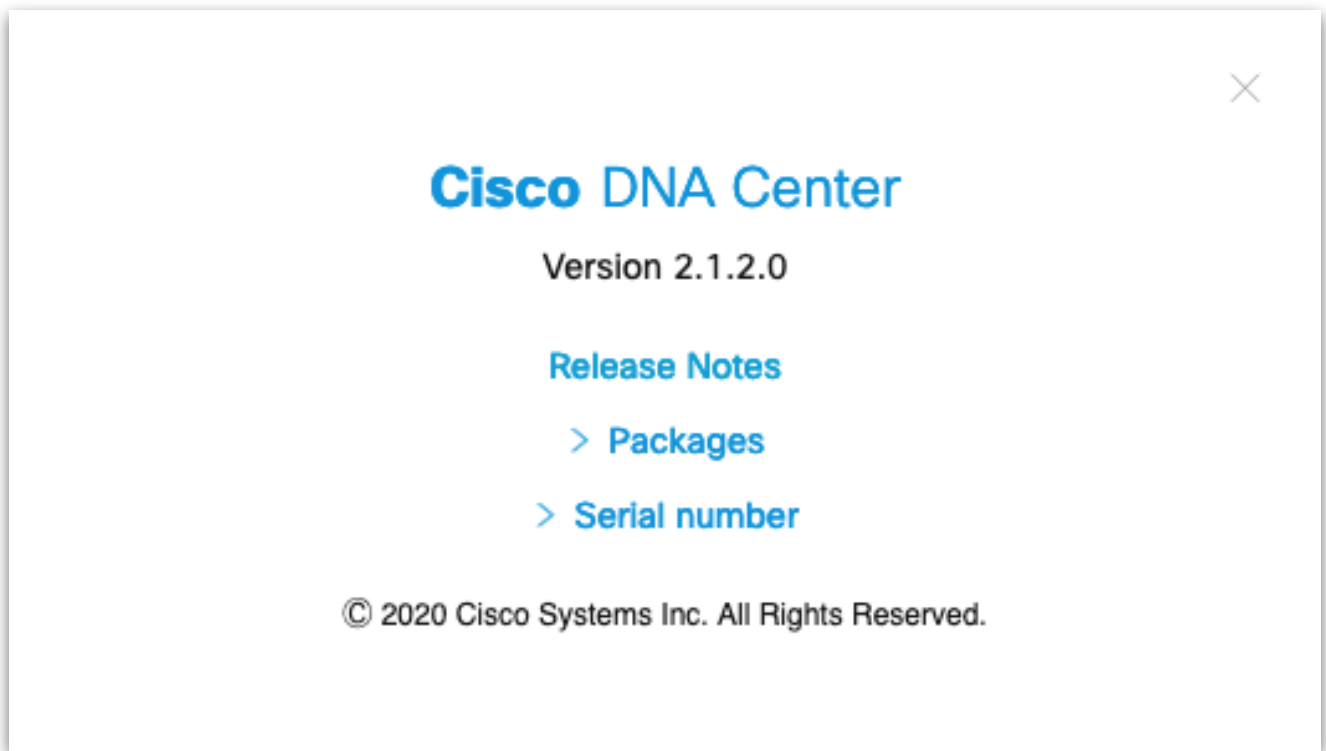
The screenshot shows the Cisco DNA Center interface for a device named 'deadbeef-1.dna.local'. At the top, there are navigation links for 'All Devices' and 'deadbeef-1.dna.local', along with a 'Run Commands' button (circled in red) and a 'View 360' link. The device status is 'Reachable' and 'Managed', with an IP address of 192.168.0.101. The device model is 'Cisco Catalyst 9200L Switch Stack' and its role is 'ACCESS'. The uptime is '33 days 12 hrs'. The site is 'Global/Deadbeef/Area-1/Building-1'. On the left, a sidebar lists 'DETAILS' categories: Interfaces, Ethernet Ports, VLANs, Hardware & Software, Configuration, Power, and Fans. The main content area displays a 'Compliance Summary' with three cards: 'STARTUP VS RUNNING CONFIGURATION' (588 lines in both), 'SOFTWARE IMAGE' (version 17.03.01...), and 'CRITICAL SECURITY ADVISORIES' (0). A 'Run Compliance' button is also visible.

This screenshot shows the same Cisco DNA Center interface as above, but with the 'Command Runner' window open. The 'Run Commands' button is circled in red, and a red arrow points to the 'Command Runner' window title bar. The window title is 'Command Runner' and the address bar shows 'deadbeef-1.dna.local@192.168.0.101'. The window content displays a welcome message: 'Welcome to Cisco DNA Center command runner. You can access this window from anywhere using the key combination Q+T. You can access recently viewed devices using the key combination Q+D. Note: You can enter "man" anytime to get the list of currently supported commands and shortcuts.' The prompt 'deadbeef-1.dna.local>' is visible at the bottom of the window.

Command Runner (cont.)

The screenshot shows the Cisco DNA Center interface for a device named 'deadbeef-1.dna.local'. The left sidebar contains navigation options: DETAILS (Interfaces, Ethernet Ports, VLANs, Hardware & Software, Configuration, Power, Fans), COMPLIANCE (Summary), and a top navigation bar with 'All Devices > deadbeef-1.dna.local', 'Run Commands', and 'View 360'. The main content area displays a 'Compliance Summary' with a 'STARTUP VS RUNNING CONFIGURATION' section showing 588 lines in both configurations. A 'Command Runner' window is open, displaying the output of the 'show run' command. The output includes the current configuration size (28541 bytes) and a list of configuration commands such as 'service timestamps debug datetime msec', 'hostname deadbeef-1', and 'vrf definition Mgmt-vrf'.

This screenshot is similar to the one above, showing the same Cisco DNA Center interface for 'deadbeef-1.dna.local'. The 'Command Runner' window now displays the output of the 'show start' command. The output shows the startup configuration size (22625 out of 2097152 bytes) and lists the configuration commands that are loaded into the startup configuration, including 'hostname deadbeef-1' and 'vrf definition Mgmt-vrf'.



The End.