

Cisco Ultra Cloud Core CPC AAA- API Reference, Release 2026.04

First Published: 2026-11-05

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

Full Cisco Trademarks with Software License ?

PREFACE

About this Guide v

About This Guide v

Audience v

Additional Support vi

Conventions (all documentation) vi

Communications, Services, and Additional Information vii

Important Notes viii

CHAPTER 1

Overview 1

General Information 1

Namespace 1

CHAPTER 2

Requests, Responses, and Schema 3

API Requests and Responses 3

Single API Request 3

CreateSubscriberRequest 3

CreateSubscriberResponse 4

GetSubscriberRequest 4

GetSubscriberResponse 5

UpdateSubscriberRequest 6

UpdateSubscriberResponse 7

DeleteSubscriberRequest 7

DeleteSubscriberResponse 8

Bulk API Request 8

CreateBulkSubscribersRequest 8
CreateBulkSubscribersResponse 10
GetSubscribersRequest 10
GetSubscribersResponse 11
UpdateSubscribersRequest 11
UpdateSubscribersResponse 13
DeleteSubscribersRequest 13
DeleteSubscribersResponse 14
Refresh profile API implementation in cnAAA 14
Delete session API implementation in cnAAA 15

CHAPTER 3 SOAP API Support 17
Feature History 17
Overview 17
Bulk Provisioning 18
Dynamic Policy Push 18



About this Guide

- About This Guide, on page v
- Audience, on page v
- Additional Support, on page vi
- Conventions (all documentation), on page vi
- Communications, Services, and Additional Information, on page vii
- Important Notes, on page viii

About This Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This document overrides the same document available in the 22.1.0. For other functionality refer to the 22.1.0 documentation at Cisco.com.

This document is a part of the Cisco Policy Suite documentation set.

Note The PATS/ATS, ANDSF, and MOG products have reached end of life and are not supported in this release. Any references to these products (specific or implied), their components or functions in this document are coincidental and are not supported.

Audience

This guide is best used by these readers:

- Network administrators

- Network engineers
- Network operators
- System administrators

This document assumes a general understanding of network architecture, configuration, and operations.

Additional Support

For further documentation and support:

- Contact your Cisco Systems, Inc. technical representative.
- Call the Cisco Systems, Inc. technical support number.
- Write to Cisco Systems, Inc. at support@cisco.com.
- Refer to support matrix at https://www.cisco.com/c/en/us/support/index.html.

Conventions (all documentation)

This document uses the following conventions.

Conventions	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
courier font	Terminal sessions and information the system displays appear in courier font.
< >	Nonprinting characters such as passwords are in angle brackets.

Conventions	Indication
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.



Note Means reader take note. Notes contain helpful suggestions or references to material not covered in the manual.



Caution Means reader be careful. In this situation, you might perform an action that could result in equipment damage or loss of data.



Warning IMPORTANT SAFETY INSTRUCTIONS.

Means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS



Note Regulatory: Provided for additional information and to comply with regulatory and customer requirements.

Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

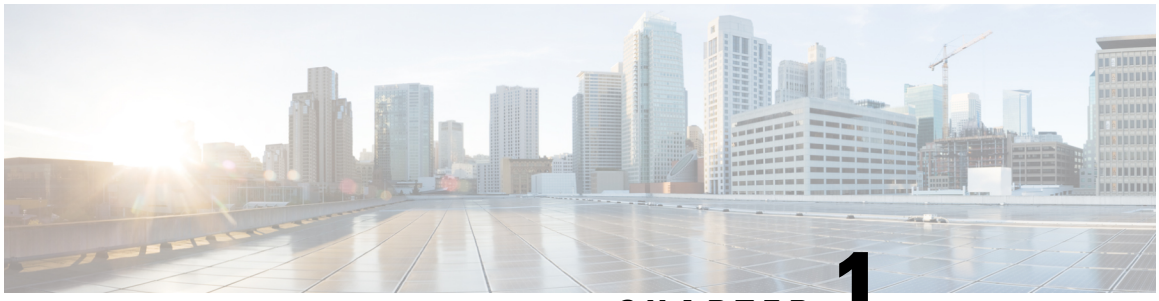
Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.

Important Notes



Important Any feature or GUI functionality that is not documented may not be supported in this release or may be customer specific, and must not be used without consulting your Cisco Account representative.



CHAPTER 1

Overview

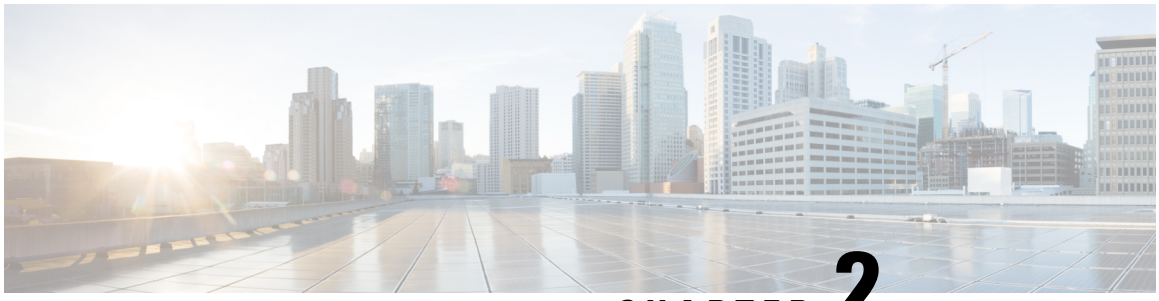
- General Information, on page 1
- Namespace, on page 1

General Information

The Unified API version is 25.1.0

Namespace

Target Namespace	http://broadhop.com/unifiedapi/soap/types
Element and Attribute Namespaces	• Global element and attribute declarations belong to this schema's target namespace. • By default, local element declarations belong to this schema's target namespace. • By default, local attribute declarations have no namespace.



CHAPTER 2

Requests, Responses, and Schema

- API Requests and Responses, on page 3

API Requests and Responses

Note Elements defined as mandatory ([1..x]) in cnAAA response schema is applicable only to success response and not to failure response.

Single API Request

CreateSubscriberRequest

This API creates a subscriber in the USuM database. It also allows provisioning of initial quota/balance using the CreateBalance element.

Schema

```
<CreateSubscriberRequest>
<audit> AuditType </audit> [0..1] ?
<subscriber> SubscriberType </subscriber> [1]
</CreateSubscriberRequest>
```

Example

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
    -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/CreateSubscriberRequest\""
 \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv=
"http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:CreateSubscriberRequest>
         <subscriber>
        <name>
            <fullName>LaasUser_41</fullName>
```

```
        </name>
        <credential>
            <networkId>uniqueIdentifier</networkId>
            <password>password</password>
        </credential>
        <service>
            <code>Plan0</code>
            <enabled>true</enabled>
        </service>
        <avp>
            <code>Called-Station-Id</code>
            <value>LaasUser_41</value>
        </avp>
        <avp>
            <code>CIRCLE_CODE</code>
            <value>WB</value>
        </avp>
        <avp>
            <code>PINCODE</code>
            <value>405041</value>
        </avp>
        <status>ACTIVE</status>
    </subscriber>
     </typ:CreateSubscriberRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

CreateSubscriberResponse

Schema

```
<CreateSubscriberResponse>
   <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
   <errorMessage> xsd:string </errorMessage> [1] ?
</CreateSubscriberResponse>
```

Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <CreateSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </CreateSubscriberResponse>
  </se:Body>
</se:Envelope>
```

GetSubscriberRequest

This API retrieves a subscriber from the USuM database.

Note **Success Response**

By default, the GetSubscriber API returns error code 15 (failure) if no subscriber is found. The `success.response.nonexistent.user` property has been introduced in Ops-Center. This property allows cnAAA can to control the behavior of the GetSubscriber API response. If the value is set to true, the GetSubscriber API returns error code 0 even when no subscriber is found.

See the QueryBalance request for information about the boolean options: `includeExpiredData` and `excludeReservationsFromCreditTotal`

Schema:

```
<GetSubscriberRequest>
<audit> AuditType </audit> [0..1] ?
<networkId> xsd:string </networkId> [1] ?
<returnSessions> xsd:boolean </returnSessions> [0..1] ?
<returnBalances> xsd:boolean </returnBalances> [0..1] ?
<includeExpiredData> xsd:boolean </includeExpiredData> [0..1] ?
<excludeReservationsFromCreditTotal> xsd:boolean </excludeReservationsFromCreditTotal>
[0..1] ?
</GetSubscriberRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
     -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/GetSubscriberRequest\"" \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:GetSubscriberRequest>
         <typ:networkId>SVI_Subscriber_3300424</typ:networkId>
      </typ:GetSubscriberRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

Note This existing `GetSubscriberRequest` is used in a new Dual-Write Proxy logic. The `UpdateSubscriber` logic is enhanced to require specific `<id>` and `<version>` injection for data consistency.

GetSubscriberResponse

Schema:

```
<GetSubscriberResponse>
<errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
<errorMessage> xsd:string </errorMessage> [1] ?
<subscriber> SubscriberType </subscriber> [0..1]
</GetSubscriberResponse>
```

Example:

```
<se:Envelope
xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"><se:Body><GetSubscriberResponse
xmlns="http://broadhop.com/unifiedapi/soap/types"><errorCode>0</errorCode><errorMessage>
Request completed
successfully</errorMessage><subscriber><id>00cd69d02edd246467628f40</id><credential>
<networkId></networkId><password></password></credential><credential><networkId>
]]></networkId><type>Called-Station-Id</type></credential><service><code>Plan0</code><enabled>true</enabled>
</service><status>ACTIVE</status><avp><code>Called-Station-Id</code><value>IN001_3300424</value></avp><avp><code>CIRCLE_CODE
</code><value>DL</value></avp><avp><code>PINCODE</code><value>395004</value></avp><avp><code>SprKeyHash</code><value>2</value>
</avp><version>0</version>
</subscriber></GetSubscriberResponse></se:Body></se:Envelope>
```

UpdateSubscriberRequest

This API updates a subscriber. An entire subscriber object must be sent. This request also updates credentials.

It also allows provisioning initial quota/balance using the CreateBalance element.



Note The proper steps to use this API:

1. Call GetSubscriber API to get the subscriber object.
2. Update returned subscriber object as required.
3. Call UpdateSubscriber API with updated subscriber object.

Schema:

```
<UpdateSubscriberRequest>
<audit> AuditType </audit> [0..1] ?
<subscriber> SubscriberType </subscriber> [1] ?
</UpdateSubscriberRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
    -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/UpdateSubscriberRequest\""
 \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:UpdateSubscriberRequest>
         <subscriber>
                  <id>001000001ca50459681090ee</id>
                  <name>
                     <fullName>uniqueidentifier</fullName>
                  </name>
                  <credential>
                     <networkId>uniqueidentifier</networkId>
                     <password>password</password>
                  </credential>
                  <service>
                     <code>Plan0-FTTX</code>
                     <enabled>true</enabled>
                  </service>
                  <status>ACTIVE</status>
                  <avp>
                     <code>CIRCLE_CODE</code>
                     <value>WB</value>
                  </avp>
                  <version>0</version>
               </subscriber>
      </typ:UpdateSubscriberRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

Note This existing `UpdateSubscriberRequest` is used in a new Dual-Write Proxy logic. The `UpdateSubscriber` logic is enhanced to require specific `<id>` and `<version>` injection for data consistency.

UpdateSubscriberResponse

Schema

```
<UpdateSubscriberResponse>
   <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
   <errorMessage> xsd:string </errorMessage> [1] ?
</UpdateSubscriberResponse>
```

Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <UpdateSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </UpdateSubscriberResponse>
  </se:Body>
</se:Envelope>
```

DeleteSubscriberRequest

This API deletes a subscriber from the USuM database.

Note **Renamed**

The API has been renamed. It is DeleteSubscriberByKey in v1.0 API.

Schema

```
<DeleteSubscriberRequest>
   <audit> AuditType </audit> [0..1] ?
   <networkId> xsd:string </networkId> [1] ?
   <hardDelete> xsd:boolean </hardDelete> [0..1] ?
</DeleteSubscriberRequest>
```

Example

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
    -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/DeleteSubscriberRequest\""
 \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:DeleteSubscriberRequest>
         <networkId>uniqueidentifier</networkId>
         <hardDelete>true</hardDelete>
```

```
      </typ:DeleteSubscriberRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

DeleteSubscriberResponse

Schema

```
<DeleteSubscriberResponse>
   <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
   <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteSubscriberResponse>
```

Example

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
  <se:Body>
    <DeleteSubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
      <errorCode>0</errorCode>
      <errorMessage>Request completed successfully</errorMessage>
    </DeleteSubscriberResponse>
  </se:Body>
</se:Envelope>
```

Bulk API Request

CreateBulkSubscribersRequest

The CreateBulkSubscribers API enables the creation of multiple subscribers in one request. It includes subscriber details like credentials and services, ensuring correct provisioning.

Schema:

```
<CreateBulkSubscribersRequest>
<audit> AuditType </audit> [0..1] ?
<subscriber> SubscriberType </subscriber> [1..1000]
</CreateBulkSubscribersRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
     -H "SOAPAction:
\"http://broadhop.com/unifiedapi/soap/types/CreateBulkSubscribersRequest\"" \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:CreateBulkSubscribersRequest>
         <subscriber>
        <name>
            <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
            <networkId>uniqueIdentifier</networkId>
            <password>password</password>
        </credential>
        <service>
            <code>Plan0</code>
```

```
            <enabled>true</enabled>
        </service>
        <avp>
            <code>Called-Station-Id</code>
            <value>LaasUser_31</value>
        </avp>
        <avp>
            <code>CIRCLE_CODE</code>
            <value>WB</value>
        </avp>
        <avp>
            <code>PINCODE</code>
            <value>405061</value>
        </avp>
        <status>ACTIVE</status>
    </subscriber>
    <subscriber>
        <name>
            <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
            <networkId>uniqueIdentifier</networkId>
            <password>password</password>
        </credential>
        <service>
            <code>Plan0</code>
            <enabled>true</enabled>
        </service>
        <avp>
            <code>Called-Station-Id</code>
            <value>LaasUser_32</value>
        </avp>
        <avp>
            <code>CIRCLE_CODE</code>
            <value>WB</value>
        </avp>
        <avp>
            <code>PINCODE</code>
            <value>405061</value>
        </avp>
        <status>ACTIVE</status>
    </subscriber>
    <subscriber>
        <name>
            <fullName>uniqueIdentifier</fullName>
        </name>
        <credential>
            <networkId>uniqueIdentifier</networkId>
            <password>password</password>
        </credential>
        <service>
            <code>Plan0</code>
            <enabled>true</enabled>
        </service>
        <avp>
            <code>Called-Station-Id</code>
            <value>LaasUser_33</value>
        </avp>
        <avp>
            <code>CIRCLE_CODE</code>
            <value>WB</value>
        </avp>
        <avp>
            <code>PINCODE</code>
```

```
            <value>405061</value>
          </avp>
          <status>ACTIVE</status>
     </subscriber>
       </typ:CreateBulkSubscribersRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

CreateBulkSubscribersResponse

Schema:

```
<CreateBulkSubscribersResponse>
<errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
<errorMessage> xsd:string </errorMessage> [1] ?
</CreateBulkSubscribersResponse>
```

Example:

```
 <se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
   <se:Body>
      <list>
         <CreateBulkSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </CreateBulkSubscribersResponse>
         <CreateBulkSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </CreateBulkSubscribersResponse>
         <CreateBulkSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </CreateBulkSubscribersResponse>
      </list>
   </se:Body>
</se:Envelope>
```

GetSubscribersRequest

The GetSubscribers API retrieves subscriber details based on specified network IDs. It allows for querying multiple subscribers in a single request to access the information.

Schema:

```
<GetSubscribersRequest>
<audit> AuditType </audit> [0..1] ?
<networkId> xsd:string </networkId> [1..1000] ?
<returnSessions> xsd:boolean </returnSessions> [0..1] ?
<includeExpiredData> xsd:boolean </includeExpiredData> [0..1] ?
<excludeReservationsFromCreditTotal> xsd:boolean </excludeReservationsFromCreditTotal>
[0..1] ?
</GetSubscribersRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
     -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/GetSubscribersRequest\""
\
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
```

```
   <soapenv:Body>
      <typ:GetSubscribersRequest>
         <typ:networkId>SVI_Subscriber_3300422</typ:networkId>
         <typ:networkId>SVI_Subscriber_3300423</typ:networkId>
         <typ:networkId>SVI_Subscriber_3300424</typ:networkId>
      </typ:GetSubscribersRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

GetSubscribersResponse

Schema:

```
<GetSubscribersResponse>
<errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
<errorMessage> xsd:string </errorMessage> [1] ?
<subscriber> SubscriberType </subscriber> [1..1000]
</GetSubscribersResponse>
```

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/"><se:Body><list>
<GetSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types"><errorCode>
0</errorCode><errorMessage>Request completed successfully</errorMessage><subscriber>
<subscriber><id>00d69c018cab7b49680ab385</id><credential><networkId>
<![CDATA[SVI_Subscriber_3300422]]></networkId><password><![CDATA[{SSHA}
Yk6GaDy0tor24AigH7tS4FirGPY/PwE/Pz91Pw==]]>
</password></credential><credential><networkId><![CDATA[INJKNORAXXXXTW0013ENBESR001_3300422]]>
</networkId><type>Called-Station-Id</type></credential><service>
<code>Plan0</code><enabled>true</enabled>
</service><status>ACTIVE</status><avp><code>Called-Station-Id</code>
<value>INJKNORAXXXXTW0013ENBESR001_3300422</value></avp><avp>
<code>CIRCLE_CODE</code><value>DL</value>
</avp><avp><code>PINCODE</code><value>385006</value></avp><avp><code>SprKeyHash</code><value>1</value>
</avp><version>0</version></subscriber><subscriber><id>00e69c018cab7b49680ab385</id><credential><networkId>
<![CDATA[SVI_Subscriber_3300423]]></networkId><password><![CDATA[{SSHA}a6CJLxyyf2h2KZmzgWxCehNixGRIV0gTP3c/Pw==]]>
</password></credential><credential><networkId><![CDATA[INJKNORAXXXXTW0013ENBESR001_3300423]]>
</networkId><type>Called-Station-Id</type></credential><service><code>Plan0</code><enabled>true</enabled></service>
<status>ACTIVE</status><avp><code>Called-Station-Id</code><value>INJKNORAXXXXTW0013ENBESR001_3300423</value></avp>
<avp><code>CIRCLE_CODE</code><value>KO</value></avp><avp><code>PINCODE</code><value>385007</value></avp><avp><code>
SprKeyHash</code><value>0</value></avp><version>0</version></subscriber><subscriber><id>00d69c01e2171601680ab385
</id><credential><networkId><![CDATA[SVI_Subscriber_3300424]]></networkId><password>
<![CDATA[{SSHA}qFkpMzgCrSyVepwSOitG5JnEMDczOj90Pz9XPw==]]></password></credential><credential><networkId>
<![CDATA[INJKNORAXXXXTW0013ENBESR001_3300424]]></networkId><type>Called-Station-Id</type></credential><service>
<code>Plan0</code><enabled>true</enabled></service><status>ACTIVE</status><avp><code>Called-Station-Id</code>
<value>INJKNORAXXXXTW0013ENBESR001_3300424</value></avp><avp><code>CIRCLE_CODE</code><value>AP</value></avp><avp>
<code>PINCODE</code><value>385008</value></avp><avp><code>SprKeyHash</code><value>2</value></avp><version>0</version>
</subscriber><subscriber><id>00f69c01e2171601680ab385</id><credential><networkId></GetSubscribersResponse></list>
</se:Body></se:Envelope>
```

UpdateSubscribersRequest

The UpdateSubscribers API modifies subscriber details, including credentials and services, for active subscribers in the system. It ensures updates are reflected in the subscriber database.

Schema:

```
<UpdateSubscribersRequest>
<audit> AuditType </audit> [0..1] ?
<subscriber> SubscriberType </subscriber> [1..1000] ?
</UpdateSubscribersRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
    -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/UpdateSubscribersRequest\""
 \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:UpdateSubscribersRequest>
         <subscriber>
                  <id>00200000602b083e680f3b6f</id>
                  <name>
                     <fullName>uniqueIdentifier</fullName>
                  </name>
                  <credential>
                     <networkId>uniqueIdentifier</networkId>
                     <password>password</password>
                  </credential>
                  <service>
                     <code>Plan0-FTTX</code>
                     <enabled>true</enabled>
                  </service>
                  <status>ACTIVE</status>
                  <avp>
                     <code>CIRCLE_CODE</code>
                     <value>MU</value>
                  </avp>
                  <avp>
                     <code>SprKeyHash</code>
                     <value>0</value>
                  </avp>
                  <version>0</version>
               </subscriber>
            <subscriber>
                  <id>00300000602b083e680f3b6f</id>
                  <name>
                     <fullName>uniqueIdentifier</fullName>
                  </name>
                  <credential>
                     <networkId>uniqueIdentifier</networkId>
                     <password>password</password>
                  </credential>
                  <service>
                     <code>Plan0-FTTX</code>
                     <enabled>true</enabled>
                  </service>
                  <status>ACTIVE</status>
                  <avp>
                     <code>CIRCLE_CODE</code>
                     <value>MU</value>
                  </avp>
                  <avp>
                     <code>SprKeyHash</code>
                     <value>1</value>
                  </avp>
                  <version>0</version>
               </subscriber>
            <subscriber>
                  <id>00100000658051666680f3bde</id>
                  <name>
                     <fullName>uniqueIdentifier</fullName>
```

```
                    </name>
                    <credential>
                       <networkId>uniqueIdentifier</networkId>
                       <password>password</password>
                    </credential>
                    <service>
                       <code>Plan0-FTTX</code>
                       <enabled>true</enabled>
                    </service>
                    <status>ACTIVE</status>
                    <avp>
                       <code>CIRCLE_CODE</code>
                       <value>MU</value>
                    </avp>
                    <avp>
                       <code>SprKeyHash</code>
                       <value>2</value>
                    </avp>
                    <version>0</version>
                 </subscriber>
                  </typ:UpdateSubscribersRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

UpdateSubscribersResponse

Schema:

```
<UpdateSubscribersResponse>
<errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
<errorMessage> xsd:string </errorMessage> [1] ?
</UpdateSubscribersResponse>
```

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
   <se:Body>
      <list>
         <UpdateSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </UpdateSubscribersResponse>
         <UpdateSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </UpdateSubscribersResponse>
         <UpdateSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </UpdateSubscribersResponse>
      </list>
   </se:Body>
</se:Envelope>
```

DeleteSubscribersRequest

The DeleteSubscribers API removes subscribers from the system based on specified network IDs. It supports hard deletions, offering flexible option for managing subscriber data.

Schema:

```
<DeleteSubscribersRequest>
<audit> AuditType </audit> [0..1] ?
```

```
<networkId> xsd:string </networkId> [1..1000] ?
<hardDelete> xsd:boolean </hardDelete> [0..1] ?
</DeleteSubscribersRequest>
```

Example:

```
curl -k -X POST "https://pcf.unified-api.203.0.113.1.nip.io/apirouter" \
     -H "Content-Type: text/xml; charset=utf-8" \
    -H "SOAPAction: \"http://broadhop.com/unifiedapi/soap/types/DeleteSubscribersRequest\""
 \
     -d '<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="http://broadhop.com/unifiedapi/soap/types">
   <soapenv:Header/>
   <soapenv:Body>
      <typ:DeleteSubscribersRequest>
         <networkId>uniqueidentifier</networkId>
         <networkId>uniqueidentifier</networkId>
         <networkId>uniqueidentifier</networkId>
         <hardDelete>true</hardDelete>
      </typ:DeleteSubscribersRequest>
   </soapenv:Body>
</soapenv:Envelope>'
```

DeleteSubscribersResponse

Schema:

```
<DeleteSubscribersResponse>
<errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
<errorMessage> xsd:string </errorMessage> [1] ?
</DeleteSubscribersResponse>
```

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
   <se:Body>
      <list>
         <DeleteSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </DeleteSubscribersResponse>
         <DeleteSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </DeleteSubscribersResponse>
         <DeleteSubscribersResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
            <errorCode>0</errorCode>
            <errorMessage>Request completed successfully</errorMessage>
         </DeleteSubscribersResponse>
      </list>
   </se:Body>
</se:Envelope>
```

Refresh profile API implementation in cnAAA

RefreshSubscriberProfileRequest

The Refresh Profile API in cnAAA accepts a mac-address. This API triggers the policy engine. The policy engine queries the SPR, fetches the subscriber record, and processes CoA if the session is available in CDL.

Schema:

```
<RefreshSubscriberProfileRequest>
 <audit> AuditType </audit> [0..1] ?
 <networkId> xsd:string </networkId> [1] ?
</RefreshSubscriberProfileRequest>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body><RefreshSubscriberProfileRequest xmlns="http://broadhop.com/unifiedapi/soap/types">

 <networkId>uniqueidentifier</networkId>
</RefreshSubscriberProfileRequest>
</se:Body>
</se:Envelope>
```

RefreshSubscriberProfileResponse

Schema:

```
<RefreshSubscriberProfileResponse>
 <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
 <errorMessage> xsd:string </errorMessage> [1] ?
</RefreshSubscriberProfileResponse>
```

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body>
<RefreshSubscriberProfileResponse>
 <errorCode>0</errorCode>
 <errorMessage>Request completed successfully</errorMessage>
 <sessionId>192.168.175.148:00012</sessionId>
 <sessionId>192.168.175.148:00013</sessionId>
 <sessionId>192.168.175.148:00014</sessionId>
</RefreshSubscriberProfileResponse>
</se:Body>
</se:Envelope>
```

Delete session API implementation in cnAAA

The Delete Session API in cnAAA accepts a mac-address. This API triggers the policy engine to delete the CDL sessions of that subscriber.

DeleteSubscriberSessionRequest

Schema:

```
<DeleteSubscriberSessionRequest>
 <networkId> xsd:string </networkId> [1] ?
 <subscriberId> xsd:string </subscriberId> [1] ?
</DeleteSubscriberSessionRequest>
```

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body>
<DeleteSubscriberSessionRequest xmlns="http://broadhop.com/unifiedapi/soap/types">
 <networkId>uniqueidentifier</networkId>
 <subscriberId>00c5000042e99b6068bed7ce</subscriberId>
</DeleteSubscriberSessionRequest>
</se:Body>
</se:Envelope>
```

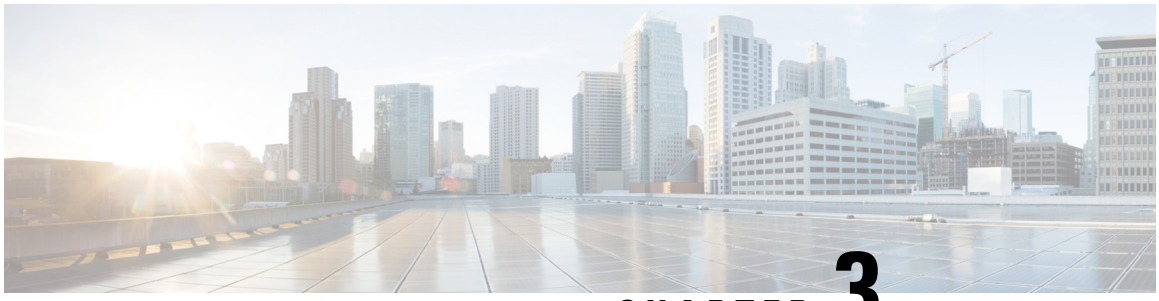
DeleteSubscriberSessionResponse

Schema:

```
<DeleteSubscriberSessionResponse>
 <errorCode> xsd:integer (0 >= value < 1000) </errorCode> [1] ?
 <errorMessage> xsd:string </errorMessage> [1] ?
</DeleteSubscriberSessionResponse>
```

Example:

```
<se:Envelope xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:Body>
<DeleteSubscriberSessionResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
<errorCode>0</errorCode>
<errorMessage>Request completed successfully</errorMessage>
<sessionCount>3</sessionCount>
<sessionId>192.168.175.148:00012</sessionId>
<sessionId>192.168.175.148:00013</sessionId>
<sessionId>192.168.175.148:00014</sessionId>
</DeleteSubscriberSessionResponse>
</se:Body>
</se:Envelope>
```



CHAPTER 3

SOAP API Support

- Feature History, on page 17
- Overview, on page 17
- Bulk Provisioning, on page 18
- Dynamic Policy Push, on page 18

Feature History

Feature Name	Release Information	Description
SOAP API Support for Subscriber CRUD Operations	2025.01	SOAP API Support for cnAAA allows for bulk management of subscriber accounts through a single API request, enabling Create, Read, Update, and Delete operations. This API also supports dynamic policy updates for active user sessions at the Broadband Network Gateway (BNG), optimizing resource management and minimizing operational redundancies.

Overview

Cloud Native Authentication, Authorization, and Accounting (cnAAA) is a system designed to manage network access and enforce security policies. It plays a crucial role in controlling network access and managing resources efficiently. cnAAA supports bulk provisioning of multiple users and allows dynamic policy updates based on real-time inputs from the Online Charging System (OCS). With a single Simple Object Access Protocol (SOAP) API request, users can perform Create, Read, Update, and Delete (CRUD) operations on multiple subscribers, a significant improvement over the previous limitation of handling one subscriber at a time.

Bulk Provisioning

Bulk provisioning refers to the ability to execute CRUD operations on multiple subscriber accounts simultaneously through a single SOAP API request. This capability minimizes redundant tasks and reduces errors in subscriber management.

Procedure

Step 1 The BNG initiates a Subscriber Provisioning SOAP request to manage subscriber data, sent to cnAAA through an HTTPS interface. The UnifiedAPI endpoint on cnAAA handles these requests.

Step 2 Based on the CRUD operation specified, cnAAA interacts with the Subscriber Profile Repository (SPR) to execute the operation.

Step 3 After executing the operation, cnAAA sends a response back to the BNG to confirm success or failure.

Dynamic Policy Push

Dynamic Policy Push updates policies for active subscriber sessions at the Broadband Network Gateway (BNG) using OCS inputs. cnAAA sends a Change of Authorization (CoA) message to the BNG, including details like the NAS-IP-Address, to ensure precise application. This process implements policy changes seamlessly without interrupting user sessions.



Note In `UpdateSubscribersRequest`, the dynamic policy push functions only when the session is available.