

Configure Radius AAA

IPWorks

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2017, 2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Prerequisites	1
1.2	Relation Information	2
2	Configuration Overview	3
3	Configuring Home AAA Server	5
3.1	Configuring AAA Server Type	5
3.2	Configuring Home AAA Server Network Relationship	6
3.3	Configuring Radius Stack Interface	8
3.4	Configuring Authentication and Authorization for Home AAA Server	8
3.5	Configuring Accounting for Home AAA Server	13
3.6	Configuring Dictionaries	18
3.7	Dynamic Authorization	20
4	Configuring Proxy AAA Server	27
4.1	Configuring AAA Server Type	28
4.2	Configuring Proxy Server Network Relationship	28
4.3	Configuring AAA Proxy Function	30
5	Configuring Home and Proxy AAA Server	35
6	Configuring IP Allocation	37
6.1	Configuring AAA Session Records	37
6.2	Configuring IP Allocation Function	38
6.3	Configuring AAA User	40
6.4	Viewing AAA IP Allocation Status	42
6.5	Force Releasing IPv4 Address and IPv6 Prefix	44
7	Configuring AAA Front End (Radius)	45
7.1	Enabling AAA Front End (Radius) Feature	45
7.2	Configuring CUDB Connection Pool	45
7.3	Configuring LDAP Dictionary	48
7.4	Configuring AAA Front End (Radius) Graceful Handling for CUDB Overload Protection	49



7.5	Configuring IPsec Tunnel for CUDB	50
7.6	Configuring AAA Front End (Radius) Counter in CUDB	50
8	Configuring Wi-Fi AAA	51
8.1	Configuring EAP Method	52
8.2	Configuring Local SS7 Parameters	53
8.3	Configuring Subscription Authorization Mode	53
8.4	Configuring CUI Switch	58
8.5	Configuring Showing IMSI in Access-Accept Package	58
8.6	Configuring Subscriber-Charging-Characteristics in Access-Accept	59
8.7	Configuring GT Convert	59
8.8	HSS Integration for Wi-Fi AAA Configuration	61
9	Configuring Accounting Forward/Mediation	63
10	Configuring MSISDN Authorization for Home AAA Server	65
10.1	Configuring MSISDN AuthSelector	65
10.2	Creating Users for MSISDN Authorization	66
10.3	Configuring Policies for Authorization	67
10.4	Configuring User Group for Authorization	67
10.5	Linking User and Policy	68
11	Radius AAA Operations	71
11.1	Restarting Radius AAA Server	71
11.2	Viewing Server Logs	71
12	Appendix A: /etc/ipworks/aaa_radius/aaa_wifi_data.xml	73
12.1	Configuration Parameters	73
12.2	Example for Wi-Fi AAA	75
	Reference List	77



1 Introduction

This document describes how to configure IPWorks Radius AAA.

1.1 Prerequisites

This section states the prerequisites that must be fulfilled.

- Intermediate Linux and UNIX skills
- Concepts, terminologies, and telecommunication abbreviations, such as TCP/IP, packet data networks, and SC/PL node.
- An Ericsson Command-Line Interface (ECLI) session in Exec mode is in progress.

1.1.1 Documents

Before starting this procedure, ensure that the following documents are available:

- For more information about the basics and concepts regarding the configuration management of IPWorks, refer to [IPWorks Configuration Management](#).
- For more information about the objects configured through IPWorks CLI (ipwcli), refer to [IPWorks AAA Parameter Description](#).
- For more information about how to use the IPWorks CLI (ipwcli), refer to [Command Line Interface User Guide for IPWorks SS](#).
- This document only introduces the most commonly used configuration scenarios, for complete information about the objects configured through ECLI, refer to [Managed Object Model \(MOM\)](#).

1.1.2 Tools

Not Applicable.

1.1.3 Conditions

Before starting this procedure, the following conditions must apply:

- IPWorks installation is completed.
- Each functionality, associated with a specific license, must be valid and running normally in the license server.



For more information about IPWorks license related information, refer to [License Management](#).

- Storage Server is started.
- Radius AAA must be initially configured.
- SS7 Stack must be configured when the following functions are used:
 - EAP-AKA
 - EPA-SIM

For more information about how to configure SS7 Stack, refer to [Configure SS7 for AAA](#).

1.2 Relation Information

Trademark information, typographic conventions, and definition and explanation of abbreviations and terminology can be found in the following documents:

- [Trademark Information](#)
- [Typographic Conventions](#)
- [Glossary of Terms and Acronyms](#)



2 Configuration Overview

This document provides the configuration procedures of Radius AAA based on the following situations:

- AAA server types
 - Section 3 on page 5
 - Section 4 on page 27
 - Section 5 on page 35
- AAA server functions
 - Section 6 on page 37
 - Section 7 on page 45
 - Section 8 on page 51
 - Section 9 on page 63
 - Section 10 on page 65
- Section 11 on page 71

Note: If **Geographic Redundancy** is enabled, some Radius AAA objects only need to be configured in one site and they will be replicated automatically to another site, while other Radius AAA objects must be configured in both sites. For more detail, refer to section Redundancy with Single Provisioning in *IPWorks Geographic Redundancy*.





3 Configuring Home AAA Server

This section describes the configuration of AAA server that works as standalone home server.

Figure 1 shows a sample network for AAA standalone home server. This sample network includes a home AAA server (10.0.0.1) and a NAS client (10.0.0.2).

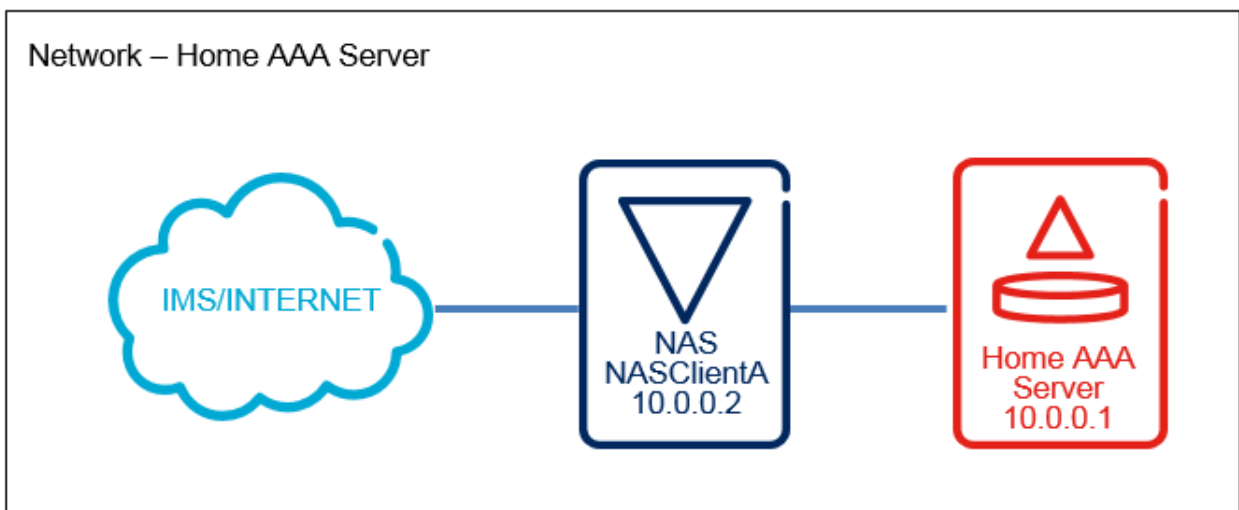


Figure 1 Sample Network 1 - Home AAA Server

The configuration of home AAA server in standalone home server includes the following topics:

- Section 3.1 Configuring AAA Server Type on page 5
- Section 3.2 Configuring Home AAA Server Network Relationship on page 6
- Section 3.3 Configuring Radius Stack Interface on page 8
- Section 3.4 Configuring Authentication and Authorization for Home AAA Server on page 8
- Section 3.5 Configuring Accounting for Home AAA Server on page 13
- Section 3.6 Configuring Dictionaries on page 18
- Section 3.7 Dynamic Authorization on page 20

3.1 Configuring AAA Server Type

Set the attribute `proxyServerType` in the MO `ProxyControl`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```



```
>configure
```

```
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,ProxyControl=1
```

```
(config-ProxyControl=1)>proxyServerType=STANDALONE_HOME_SERVER
```

```
(config-ProxyControl=1)>commit
```

```
(ProxyControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.2 Configuring Home AAA Server Network Relationship

To configure network relationship between AAA server and the relevant clients, perform the following procedures:

- Section 3.2.1 Creating Home AAA Server on page 6
- Section 3.2.2 Creating Clients for Home AAA Server on page 7

3.2.1 Creating Home AAA Server

In Sample Network 1, home AAA server is configured for functions including AAA and Dynamic Authorization (DA).

1. Start the IPWorks CLI on the Storage Server.

```
#ipwcli
```

2. Create AAA server objects.

Command Syntax:

```
IPWorks> create AAAServer -set name=<AAA Server Name>;address=<PL OAM Address>
```

For example:

```
IPWorks> create AAAServer -set name=aaasrv1;address=169.254.100.3
```

```
1 object(s) created.
```

```
IPWorks> create AAAServer -set name=aaasrv2;address=169.254.100.4
```

```
1 object(s) created.
```

3. Display the newly created AAA server objects.



Command Syntax:

```
IPWorks> list AAAserver <AAA Server Name>
```

For example:

```
IPWorks> list AAAserver aaasrv1
```

```
[AAAServer aaasrv1]
  Name: aaasrv1
  Address: 169.254.100.3
```

```
IPWorks> list AAAserver aaasrv2
```

```
[AAAServer aaasrv2]
  Name: aaasrv2
  Address: 169.254.100.4
```

3.2.2 Creating Clients for Home AAA Server

In Sample Network 1, the configured Home AAA server has one client with the IP address of 10.0.0.2.

Shared secret should be configured for the secure communication between the NAS Client and Home AAA Server.

Both IPv4 and IPv6 are supported to work with CUDB.

1. Create an object instance of ClientSharedSecret in the MO ClientSharedSecretMgr.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)> dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusStack=1,SharedSecretMgr=1,ClientSharedSecretMgr=1
(config-ClientSharedSecretMgr=1)> ClientSharedSecret=1
```

2. Set the IP address of the client.

```
(config-ClientSharedSecret=1)> clientIPAddr="10.0.0.2"
```

3. Configure the Shared secret of the client.

```
(config-ClientSharedSecret=1)> sharedSecretValue="AAA-Sharedsecret"
(config-ClientSharedSecret=1)> type=ALL
(config-ClientSharedSecret=1)> commit
(ClientSharedSecret=1)>exit
```

Note: The configuration takes effect immediately.



3.3 Configuring Radius Stack Interface

1. Configure the traffic IP and port for Authentication and Authorization message.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusStack=1,RadiusInterface=1
(RadiusInterface=1)>configure
(config-RadiusInterface=1)> authAuthzAddress="10.0.0.1"
(config-RadiusInterface=1)> authAuthzPort=1812
```

2. Configure the traffic IP and port for Accounting message.

```
(config-RadiusInterface=1)> acctAddress="10.0.0.1"
(config-RadiusInterface=1)> acctPort=1813
```

3. Configure the IP type.

```
(config-RadiusInterface=1)>localhostBindIPType=IPv4
(config-RadiusInterface=1)> commit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.4 Configuring Authentication and Authorization for Home AAA Server

The configuration of authentication and authorization for home AAA server includes the following topics:

- Section 3.4.1 Configuring Authentication and Authorization Selector on page 8
- The following topics are performed when CUDB is not used:
 - Section 3.4.2 Creating Users for Authentication and Authorization (PAP and CHAP) on page 9
 - Section 3.4.3 Configuring Policies for Authorization on page 10
 - Section 3.4.4 Configuring User Group for Authorization on page 11
 - Section 3.4.5 Linking User and Policy on page 12
- Section 3.4.6 Stripping Local Realm on page 13

3.4.1 Configuring Authentication and Authorization Selector

Some authentication and authorization selector have already been configured when IPWorks Radius AAA installation by default. The Radius AAA uses the selector to choose a satisfied function module that should respond and handle the coming request.



Following is an example to show the configuration of selector in the MO
AuthSelectorMgr:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=
1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,AuthMethodControl=1
,AuthSelectorMgr=1

(AuthSelectorMgr=1)>show -v --recursive

AuthSelectorMgr=1
  authSelectorMgrId="1"
  AuthSelector=a3eap
    authSelectorId="a3eap"
    isNeedAuthorization=false
    rank=10
    triggerAVPs
      "EAP-Message = *"
      "User-Name = *"
  AuthSelector=authorization
    authSelectorId="authorization"
    isNeedAuthorization=false
    rank=40
    triggerAVPs <default>
      "User-Name = *"
  AuthSelector=pap
    authSelectorId="pap"
    isNeedAuthorization=true
    rank=20
    triggerAVPs
      "User-Password = *"
      "User-Name = *"
  AuthSelector=chap
    authSelectorId="chap"
    isNeedAuthorization=true
    rank=30
    triggerAVPs
      "CHAP-Password = *"
      "User-Name = *"
```

Note: The configuration takes effect after Radius AAA server restarts.

3.4.2 Creating Users for Authentication and Authorization (PAP and CHAP)

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

The parameter username has the following meanings:

— The username of the user.



- The access loop ID (Access Circuit ID) value for Fixed Access IPoE user.

Note: AAA is expected to receive the Access loop ID in AVP User-Name.

For more information, refer to the Section AAAUser in IPWorks AAA Parameter Description.

The ways to create the users are same for different user.

Create users according to the following examples through IPWorks CLI:

1. Execute the following command.

```
IPWorks> create AAAUser -set username=<username>;password=123456789A123456789A123456789A12
```

For example:

```
IPWorks> create AAAUser -set username=AAA-Test;password=123456789A123456789A123456789A12
```

```
1 object(s) created.
```

2. Confirm that the new users are created successfully.

```
IPWorks> list AAAUser
```

```
[AAAUser aaa-test]
  Username: AAA-Test
  Password: *****
[AAAUser ims-user]
  Username: IMS-User
  Password: *****
[AAAUser ipoe-user]
  Username: IPoE-User
  Password: *****
```

3.4.3 Configuring Policies for Authorization

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

1. Create two policies.

```
IPWorks> create AAPolicy -set name=policy1;checklist="User-Name=IMS-User";replylist="User-Name=$REQUEST"
```

```
1 object(s) created.
```

```
IPWorks> create AAPolicy -set name=policy2;checklist="(User-Name?1 && User-Name!=IMS-User)";replylist="User-Name=$REQUEST,Login-IP-Host = \"10.170.4.169\""
```



- 1 object(s) created.
2. Display the newly created policies.

```
IPWorks> list AAAPolicy
```

```
[AAAPolicy policy1]
  Name: policy1
  Checklist: User-Name = "IMS-User"
  Replylist: User-Name = $REQUEST
[AAAPolicy policy2]
  Name: policy2
  Checklist: ( User-Name ? 1 && User-Name != "IMS-User" )
  Replylist: User-Name = $REQUEST, Login-IP-Host = 10.170.4.169
```

The checklist of policy1 is used to check the incoming packet AVPs, and to ensure that the value of attribute User-Name is IMS-User. If the check result is true, the AVPs in replylist will be appended to the response packet. User-Name with the value in the request packet will be appended.

The checklist of policy2 is used to check the incoming packet AVPs, and to ensure that attribute User-Name appears once in the incoming package and its value is not IMS-User. If the check result is true, the AVPs in replylist will be appended to the response packet. User-Name with the value in the request packet and Login-IP-Host with the value of 10.170.4.169 will be appended.

3. Modify the newly created policies.

```
IPWorks> modify AAAPolicy policy1 -set checklist=<"User-Name=IMS-User">;replylist=<"User-Name=$REQUEST">
```

3.4.4 Configuring User Group for Authorization

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

1. Create a user group named IMS-Group.

```
IPWorks> create AAAUserGroup -set name=IMS-Group
```

```
1 object(s) created.
```

2. Modify the newly created user group to set policies policy1 and policy2.

```
IPWorks> modify AAAUserGroup IMS-Group -set policy=policy1,policy2
```

```
Working on 1 object(s).
```

3. Display the newly created user group.

```
IPWorks> list AAAUserGroup
```

```
[AAAUserGroup ims-group]
  Name: IMS-Group
  Policy: policy1, policy2
```



3.4.5 Linking User and Policy

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

The AAA user AAA-Test is directly linked to the policies. The policies will be checked for the user in the alphabetical sequence in attribute policy. The first policy checked with TRUE is selected for authentication or authorization.

1. Modify the AAA user to set the policies.

```
IPWorks> modify AAAUser AAA-Test -set policy=policy1,policy2
```

```
Working on 1 object(s).  
1 object(s) were updated.
```

2. Display the modified AAA user AAA-Test.

```
IPWorks> list AAAUser AAA-Test
```

```
[AAAUser aaa-test]  
Username: AAA-Test  
Password: *****  
Policy: policy1, policy2
```

3. The AAA user IMS-User is indirectly linked to policy through the user group. The policies will be checked for the user in the alphabetical sequence in attribute policy of user group IMS-Group. The first policy checked with TRUE is selected for authentication or authorization.

```
IPWorks> modify AAAUser IMS-User -set groupname=IMS-Group
```

```
Working on 1 object(s).  
1 object(s) were updated.
```

4. Display the modified AAA user IMS-User.

```
IPWorks> list AAAUser IMS-User
```

```
[AAAUser ims-user]  
Username: IMS-User  
Password: *****  
GroupName: IMS-Group
```

If the user is directly linked to the policy, that is, attribute policy of the user is set, the policy of the user group will not impact AAA user policy selection



strategy. When there are multiple directly linked policies for the user, the first policy whose checklist is evaluated as TRUE will be selected.

If the user is indirectly linked to the policy through the user group, the policy selection strategy is the same as that for the directly linked policy in one user group, that means the first policy whose checklist is evaluated as TRUE will be selected.

Note: It is strongly recommended to set policies mutually exclusive for one user or user group.

3.4.6 Stripping Local Realm

When the user's realm has to be stripped for local authentication and authorization, the following AAArealm needs to be configured. If attribute Striprealm is set to true, then in Access-Request messages sent to local home AAA server, the realm of User-Name will be stripped.

Note: If the users want to realize the local authentication, must configure both authdest and acctdest as local.

1. Create the stripped AAA realm for local authentication and authorization.

For example:

```
IPWorks> create AAArealm ericsson.com -set striprealm=true;authdest=local;acctdest=local
```

```
1 object(s) created.
```

2. Make the creation take effect.

```
IPWorks> update aaaserver
```

3. Restart Radius AAA server.

See Section 11.1 Restarting Radius AAA Server on page 71 for details.

4. Display the AAA realm.

```
IPWorks> list
```

```
[AAAR realm ericsson.com]
  Name: ericsson.com
  StripRealm: true
  AuthDest: local
  AcctDest: local
```

3.5 Configuring Accounting for Home AAA Server

The configuration of accounting for home AAA server includes the following topics:



— Section 3.5.1 Configuring CSV Record on page 14

— Choose one of the configuration according to the accounting types:

Note: When accounting function is used, one or more messages are exchanged:

For the accounting start and stop, only one message is exchanged.

For the accounting update, many messages are exchanged.

- Section 3.5.2 Message-based Accounting Configuration on page 14
- Section 3.5.3 Session-based Accounting Configuration on page 16

3.5.1 Configuring CSV Record

A default CSV record is set as follows. It will control accounting messages content recording and determine what AVPs will be recorded in CSV files.

1. Display details of the CSV record.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AccountingService=1
(AccountingService=1)>show -v
AccountingService=1
  accountingServiceId="1"
  csvMode=MESSAGE_BASED_CSV <default>
  csvRecords="User-Name,NAS-IP-Address,NAS-Port,Acct-Status-Type,Acct-Session-Id"
  AcctFinder=1
  AcctForward=1
  CSVEngineCommon=1
  CSVFTPInformation=1
  CsvGenerateMethod=1
```

2. Modify the CSV record.

```
(AccountingService=1)>configure
(config-AccountingService=1)> csvRecords="User-Name,NAS-IP-Address,
NAS-Port,Acct-Status-Type,Acct-Session-Id,Called-Station-Id,Calling-Station-Id"
(config-AccountingService=1)> commit
(config-AccountingService=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.5.2 Message-based Accounting Configuration

Message-based Accounting records all the messages in sequence. In this case, it might have many records for each session in CSV files.

The user can also transfer the CSV files to a remote FTP server.

Message-based Accounting can be activated by set attribute `csvMode` to `MESSAGE_BASED_CSV` in MO `AccountingService`.



```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1
,IPWorksRadiusAAARoot=1,RadiusAAAService=1,AccountingService=1

(AccountingService=1)>configure

(config-AccountingService=1)> csvMode=MESSAGE_BASED_CSV

(config-AccountingService=1)> commit

(AccountingService=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.5.2.1 Configuring Accounting CSV Engine

A default CSV engine is configured through MO `CsvGenerateMethod`. This MO controls CSV files size, path and on which IP/Port CSV engine will listen. An example with default value is shown as follows:

```
(CsvGenerateMethod=1)>show -v

CsvGenerateMethod=1
  acctMessageReadable=false <default>
  csvBackupFileDir="/cluster/ipworks/cdr/csvbackup/"
  csvFileGenerateRule=FILE_SIZE <default>
  csvFileRecordNum=2500 <default>
  csvFileSize=5 <default>
  csvFileTimeInterval=15 <default>
  csvGenerateMethodId="1"
  csvGeneratorFileDir="/cluster/ipworks/cdr/payment/"
  csvPaymentBackupExpireTime=10080 <default>
  csvPaymentTempFileDir="/cluster/ipworks/cdr/temp/"
  generatorEngineAddress="127.0.0.1" <default>
  generatorEnginePort=56165 <default>
  pkMaxFileRecordNum=0 <default>
  pkMaxFileSize=2 <default>
```

Note: The configuration takes effect after Radius AAA server restarts.

3.5.2.2 Configuring FTP Server

IPWorks enables the user to transfer the CSV files to a remote FTP server if needed. It strengthens the data security. When the function is enabled, the CSV files is copied, instead of being moved, from the CSV Generator File Directory on SC node to the backup file directory specified in the ECIM. The CSV files located in the backup directory will be deleted expiration interval.

1. Configure the FTP Server address and port.



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AccountingService=1
(CSVFTPInformation=1)>configure
(config-CSVFTPInformation=1)>ftpServerAddress="<IP Address>"
(config-CSVFTPInformation=1)> ftpServerPort=<port>
```

Note: Modify the attribute `ftpServerAddress` and `ftpServerPort` based on customer environment.

2. Configure the user name and password of FTP server.

```
(config-CSVFTPInformation=1)> username="admin"
(config-CSVFTPInformation=1)> password="admin"
(config-CSVFTPInformation=1)> commit
```

Note: The configuration takes effect after Radius AAA restarts.

3. Display information of CSV FTP.

```
(CSVFTPInformation=1)>show -v
CSVFTPInformation=1
csvFtpInformationId="1"
ftpServerAddress="127.0.0.1" <default>
ftpServerPort=21 <default>
mode=PULL_METHOD <default>
password="1:0Ng2+qpaqA0cJKzrCPBzb6VCC+5JjKWJ"
transferFileNum=3 <default>
username="admin" <default>
```

3.5.3

Session-based Accounting Configuration

Note: Make sure that `csvengine` is configured and started if this function is enabled.

Session-based Accounting associates all the messages. In this case, it has only one record for each session in CSV files.

The user can also transfer the CSV files to a remote FTP server.

Session-based Accounting can be activated by set attribute `csvMode` to `SESSION_BASED_CSV` in `MO AccountingService`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
,IPWorksRadiusAAARoot=1,RadiusAAAService=1,AccountingService=1

(AccountingService=1)>configure

(config-AccountingService=1)> csvMode=SESSION_BASED_CSV

(config-AccountingService=1)> commit
```

Note: The configuration takes effect after Radius AAA restarts.



```
(AccountingService=1)>exit
```

3.5.3.1 Configuring Accounting CSV Engine

A default CSV engine is configured through MO `CsvGenerateMethod`. This MO controls CSV files size, path and on which IP/Port CSV engine will listen. An example with default value is shown as follows:

```
(CsvGenerateMethod=1)>show -v
```

```
CsvGenerateMethod=1
  acctMessageReadable=false <default>
  csvBackupFileDir="/cluster/ipworks/cdr/csvbackup/"
  csvFileGenerateRule=FILE_SIZE <default>
  csvFileRecordNum=2500 <default>
  csvFileSize=5 <default>
  csvFileTimeInterval=15 <default>
  csvGenerateMethodId="1"
  csvGeneratorFileDir="/cluster/ipworks/cdr/payment/"
  csvPaymentBackupExpireTime=10080 <default>
  csvPaymentTempFileDir="/cluster/ipworks/cdr/temp/"
  generatorEngineAddress="127.0.0.1" <default>
  generatorEnginePort=56165 <default>
  pkMaxFileRecordNum=0 <default>
  pkMaxFileSize=2 <default>
```

Modify the attribute `generatorEngineAddress` based on customer's actual environment.

Note: The configuration takes effect after Radius AAA restarts.

3.5.3.2 Configure FTP Server

See Section 3.5.2.2 Configuring FTP Server on page 15.

3.5.3.3 Configuring Multiple Accounting Sessions Support

IPWorks AAA supports multiple accounting sessions belonging to the same user session. Two AVPs `Acct-Multi-Session-Id` and `Class` are used to link all the accounting sessions together to the user session. To determine which AVP will be used in prior, attribute `multiSessionClassPrior` in MO `AAASessionControl` should be configured. The default value of the attribute is `false`, then `Acct-Multi-Session-Id` is used by default.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1
,IPWorksRadiusAAARoot=1,RadiusAAAService=1,AAASessionControl=1
```

```
(AAASessionControl=1)>configure
```



```
(config-AAASessionControl=1)>multiSessionClassPrior=false  
(config-AAASessionControl=1)>commit  
(AAASessionControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.6 Configuring Dictionaries

This section guides how to configure AVPs as listed below:

- Section 3.6.1 Configuring AVP Dictionary on page 18

IPWorks Radius AAA uses a group of dictionary files to verify the incoming Radius message and AVPs. You can modify the dictionary files to meet the actual requirement.

- Section 3.6.2 Configuring Vendor-Specific AVPs on page 20

IPWorks AAA supports dynamic AVP configuration to allow users to add any vendor-specific AVP based on their needs.

Prerequisites:

Users must understand the following requirements before they perform the configuration:

- The length of all Vendor-specific Attributes must not be longer than 6,144 bytes.
 - Note:** If longer than 6,144 bytes, only the attributes within the 6144 bytes are saved, and the rest are dropped.
- The sequence VSAs saved in the CSV record is the same as how users configure them.
- Any comma (",") in any single VSA is replaced with semicolon (";").

3.6.1 Configuring AVP Dictionary

The AVP dictionary is a group of XML files including one root dictionary and some AVP dictionary files. The dictionaries and files are shared in the cluster. You can configure the dictionaries in the directory `/etc/ipworks/aaa_radius/dict/`.

3.6.1.1 Configuring Root Dictionary

Root AVP dictionary (`/etc/ipworks/aaa_radius/dict/dictionary.xml`) is used as an entrance to tell IPWorks Radius AAA which dictionary file should be loaded. You can customize the dictionary file and add it into the root AVP dictionary according to the following format:



```
<dictionary-configuration>
  <dict-types>
    <dict-type value="string" />
    <dict-type value="integer" />
    <dict-type value="ipaddr" />
    <dict-type value="ipv6addr"/>
    <dict-type value="date" />
    <dict-type value="time" />
  </dict-types>

  <dictionary include="dict-nonvendor.xml" />
  <dictionary include="dict-3gpp.xml" />
  <dictionary include="dict-ericsson.xml" />
  <dictionary include="dict-ms.xml" />
  <dictionary include="dict-cisco.xml" />
  <dictionary include="dict-redback.xml" />
  <dictionary include="dict-bbf.xml" />
  <dictionary include="dict-ipworks-internal.xml" />
  <dictionary include="dict-customized.xml" />
</dictionary-configuration>
```

Note: The configuration takes effect after Radius AAA server restarts.

3.6.1.2

Configuring AVP Dictionary

IPWorks Radius AAA uses some XML files (dict-xxx.xml) to record the supported general AVPs or vendor specific attributes. You can customize and adjust the AVP definition according to the following format:

```
<avps>
  <vendor code="2352" name="Ericsson-Redback">
    <avp code="1" name="RB-Client-DNS-Pri" type="ipaddr">
      <rad-validation access-accept="*" access-accept-replylist="*"
access-challenge="*" access-reject="*" access-req="0" access-req-checklist="0"
acct-record-type="*" acct-req="*" acct-req-off="*" acct-req-on="*" acct-req-start="*"
acct-req-stop="*" acct-req-update="*" acct-response="*" avp-src="rad_attr" coa-ack="*"
coa-nak="*" coa-req="*" dm-ack="*" dm-nak="*" dm-req="*" />
    </avp>
    <avp code="2" name="RB-Client-DNS-Sec" type="ipaddr">
      <rad-validation access-accept="*" access-accept-replylist="*"
access-challenge="*" access-reject="*" access-req="0" access-req-checklist="0"
acct-record-type="*" acct-req="*" acct-req-off="*" acct-req-on="*" acct-req-start="*"
acct-req-stop="*" acct-req-update="*" acct-response="*" avp-src="rad_attr"
coa-ack="*" coa-nak="*" coa-req="*" dm-ack="*" dm-nak="*" dm-req="*" />
    </avp>
    <avp code="3" name="RB-DHCP-Max-Leases" type="integer">
      <rad-validation min-value="1" max-value="255" access-accept="*"
access-accept-replylist="*" access-challenge="*" access-reject="*" access-req="0"
access-req-checklist="0" acct-record-type="*" acct-req="*" acct-req-off="*" acct-req-on="*"
acct-req-start="*" acct-req-stop="*" acct-req-update="*" acct-response="*"
avp-src="rad_attr" coa-ack="*" coa-nak="*" coa-req="*" dm-ack="*" dm-nak="*" dm-req="*" />
    </avp>
    .
    .
    .
  </vendor>
</avps>
```

Note: The configuration takes effect after Radius AAA server restarts.



3.6.2 Configuring Vendor-Specific AVPs

To configure the vendor-specific AVP:

1. Modify `dict-customized.xml` in `/etc/ipworks/aaa_radius/dict`.

```
# cd /etc/ipworks/aaa_radius/dict
# vi dict-customized.xml
```

Save the file and exit.

2. Restart Radius AAA server.

See Section 11.1 Restarting Radius AAA Server on page 71 for details.

3.7 Dynamic Authorization

The configuration of dynamic authorization proxy includes the following topics:

- Section 3.7.1 Configuring Session Working Mode on page 20
- Section 3.7.2 CoA Message Function on page 22
- Section 3.7.3 DM Function on page 23
- Section 3.7.4 Using Searching Conditions to Send Messages on page 25
- Section 3.7.5 Dynamic Authorization Message Proxy on page 25

3.7.1 Configuring Session Working Mode

Two working modes are available.

- **NAS and Session Identifier Correlation Working Mode** uses `NASId` and Session Identifier (`Acct-Session-Id`) uniquely to identify an AAA session. In this mode, only the Access-Request message with both `NASId` and Session Identifier will be accepted. The Access-Request message with only `NASId` will be rejected with error information.

For more information about `NASId`, refer to Section `AAASession` in [IPWorks AAA Parameter Description](#).

- **NAS and Class Correlation Working Mode** uses `NASId` and `Class` uniquely to identify an AAA session. In this mode, the Access-Request message with only `NASId` is accepted, and a `Class` with a unique value will be sent back in the reply list. If the `Class` with the unique value is not included in the Accounting-Start request message, the message will be rejected with error information.

Following example shows details of a default session record. The session record controls authentication and accounting message content recording and



determines what AVPs will be recorded in a session. The configuration is for **NAS and Session Identifier Correlation Working Mode**.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AAASessionControl=1
(AAASessionControl=1)>show -v
AAASessionControl=1
  aaaSessionControlId="1"
  acctOnOffCloseAll=true <default>
  createProxySession=true <default>
  expireDMNotify=false <default>
  expireTime=1440 <default>
  multiSessionClassPrior=false <default>
  records="User-Name,NAS-Port,Framed-IP-Address,Called-Station-Id,⇒
Calling-Station-Id,Acct-Multi-Session-Id" <default>
  terminateExpireSession=true <default>
  threshold4SessionCapacity="80,90" <default>
  updateIPViaAcct=false <default>
```

More information about the session record:

- Once session configuration is changed, all the previous session data will be useless. The administrator has to clean all the session data manually.
- Session key attributes, NAS-IP-Address, NAS-Identifier, NAS-IPv6-Address, and Acct-Session-Id, are not allowed to be configured in the Records attribute.
- Other session identification attributes are allowed to be configured in the Records attribute, such as User-Name, NAS-Port, Framed-IP-Address, Called-Station-Id, Calling-Station-Id, Acct-Multi-Session-Id, NAS-Port-Id, Chargeable-User-Identity, Framed-Interface-Id, and Framed-IPv6-Prefix.

Note: The configuration takes effect after Radius AAA restarts.

3.7.1.1

Changing to NAS and Session Identifier Correlation Working Mode

1. Display information of records in the MO `AAASessionControl`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AAASessionControl=1
(AAASessionControl=1)>show -v
AAASessionControl=1
  aaaSessionControlId="1"
  acctOnOffCloseAll=true <default>
  createProxySession=true <default>
  expireDMNotify=false <default>
  expireTime=1440 <default>
  multiSessionClassPrior=false <default>
  records="Class,User-Name,NAS-Port,Framed-IP-Address,⇒
Called-Station-Id,Calling-Station-Id,Acct-Multi-Session-Id" <default>
  terminateExpireSession=true <default>
  threshold4SessionCapacity="80,90" <default>
```



```
updateIPViaAcct=false <default>
```

2. Remove Class from attribute records.

```
(AAASessionControl=1)>configure
(config-AAASessionControl=1)>records="User-Name,NAS-Port,Framed-IP-Address,
Called-Station-Id,Calling-Station-Id,Acct-Multi-Session-Id"
(config-AAASessionControl=1)>commit
(AAASessionControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.7.1.2

Changing to NAS and Class Correlation Working Mode

1. Display information of records in the MO `AAASessionControl`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AAASessionControl=1
(AAASessionControl=1)>show -v
AAASessionControl=1
  aaaSessionControlId="1"
  acctOnOffCloseAll=true <default>
  createProxySession=true <default>
  expireDMNotify=false <default>
  expireTime=1440 <default>
  multiSessionClassPrior=false <default>
  records=" User-Name,NAS-Port,Framed-IP-Address,Called-Station-Id,⇒
Calling-Station-Id,Acct-Multi-Session-Id" <default>
  terminateExpireSession=true <default>
  threshold4SessionCapacity="80,90" <default>
  updateIPViaAcct=false <default>
```

2. Add Class to attribute records.

```
(AAASessionControl=1)>configure
(config-AAASessionControl=1)> records="Class,User-Name,NAS-Port,Framed-IP-A
Called-Station-Id,Calling-Station-Id,Acct-Multi-Session-Id"
(config-AAASessionControl=1)>commit
(AAASessionControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

3.7.2

CoA Message Function

This section guides how to use the Change of Authorization (CoA) message function. For the allowed attributes for the format filed, refer to the MO `CoaFormat` in Managed Object Model (MOM).

1. Create one AAA CoA message.

For example:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,DMCOAService=1,CoaFormatMgr=1
(CoaFormatMgr=1)>configure
(config-CoaFormatMgr=1)>CoaFormat=2
(config-CoaFormat=2)>format="Acct-Session-Id=$SESSION,Login-Service=Telnet"
(config-CoaFormat=2)>name="COA2"
```



```
(config-CoaFormat=2)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

2. Display the created AAA CoA message.

For example:

```
(CoaFormat=2)>show -v
CoaFormat=2
  coaFormatId="2"
  format="Acct-Session-Id=$SESSION,Login-Service=Telnet"
  name="COA2"
(CoaFormat=2)>
(CoaFormat=2)>up
(CoaFormatMgr=1)>show -v
CoaFormatMgr=1
  coaFormatMgrId="1"
  CoaFormat=2
  CoaFormat=1
```

3. Get a Unique Session ID for sending CoA message.

For example:

```
IPWorks> list aaasession
[AAASession aaaserver1]
  UniqueSessionId: sess21338053354linux-12013524
  NasIpAddr: 10.170.4.32
  NasId: 1234
  NasType: nas-ip-address
  AcctSessionId: aaasession1
  StartTime: 2012:05:26:13:29:14
  Status: active
  UserName: aaa-test
  NasPort: 8000
  FramedIpAddress: 10.170.4.35
  LastUpdateTime: 2012:05:26:13:29:15
```

4. Send CoA message to the user with the Unique Session ID according to the configured CoA message format.

For example:

```
IPWorks> send aaaserver -message=coa -format=COA2 =>
-SessionID=sess21338053354linux-12013524

1 request messages send out.
1 ACK messages received and 0 NAK message received.
```

3.7.3

DM Function

This section guides how to use the Disconnect Message (DM) function. For the allowed attributes for the format filed, refer to the MO *DmFormat* in Managed Object Model (MOM).

1. Create AAA DM.

For example:



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,DMCOAService=1,DmFormatMgr=1
(DmFormatMgr=1)>configure
(config-DmFormatMgr=1)>DmFormat=3
(config-DmFormat=3)>format="Acct-Session-Id=$SESSION,User-Name=AAA-Test"
(config-DmFormat=3)>name="DM3"
(config-DmFormat=3)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

2. Display AAA DM.

For example:

```
(DmFormat=3)>show -v
DmFormat=3
  DmFormatId="3"
  format="Acct-Session-Id=$SESSION,User-Name=AAA-Test"
  name="DM3"
(DmFormat=3)>
(DmFormat=3)>up
(DmFormatMgr=1)>show -v
DmFormatMgr=1
  DmFormatMgrId="1"
  DmFormat=3
  DmFormat=2
  DmFormat=1
```

3. Get Unique Session ID.

For example:

```
IPWorks> list aaasession

[AAASession aaaserver1]
UniqueSessionId: sess21338053354linux-12013524
NasIpAddr: 10.170.4.32
NasId: 1234
NasType: nas-ip-address
AcctSessionId: aaasession1
StartTime: 2012:05:26:13:29:14
Status: active
UserName: aaa-test
NasPort: 8000
FramedIpAddress: 10.170.4.35
LastUpdateTime: 2012:05:26:13:29:15
```

4. Send DM to users.

— Delete the session with specified SessionID.

For example:

```
IPWorks> send aaaserver -message=dm -format=DM3 =>
-SessionID=sess21338053354linux-12013524
```

1 request messages send out.

1 ACK messages received and 0 NAK message received.



- Delete all sessions for one specified user.

For example:

```
(DmFormatMgr=1)>configure
(config-DmFormatMgr=1)>DmFormat=4
(config-DmFormat=3)>format="User-Name=AAA-Test"
(config-DmFormat=3)>name="DM4"
(config-DmFormat=3)>commit
```

```
IPWorks> send aaaserver -message=dm -format=DM4 =>
-UserName=AAA-Test
```

3.7.4 Using Searching Conditions to Send Messages

When sending CoA and disconnect request message, besides using session ID, you can also use other conditions to find the relative session. Here's an example:

```
IPWorks> send aaaserver -message=coa -format=COA1 -username=user1
-status=init -nasid=10.170.0.1
```

Where:

-username, -status and -nasid options are used as the searching conditions. AAA server will find the sessions matching the conditions and then send out CoA or disconnect request messages for the sessions. By doing so, more than one request messages can be sent out when more than one sessions match the searching condition.

3.7.5 Dynamic Authorization Message Proxy

If AAA server acts as a proxy server, it can forward DA messages and the destination is fetched from the session records. If no records are found, CoA or disconnect NAK message will be responded.





4 Configuring Proxy AAA Server

This section describes configuration of AAA server that works as standalone proxy server.

The IPWorks AAA server can act as a Radius proxy server between a Radius client (NAS) and the home AAA server. It forwards the authentication, authorization, accounting requests from Radius clients to the target home AAA servers based on the realm, and forwards or Change-of-Authorizations (CoA), and Disconnect Messages from the home AAA servers to the corresponding Radius Clients (NAS).

Figure 2 shows a sample network for AAA standalone proxy server. This sample network includes a proxy AAA server (12.0.0.1) and a NAS client (NASClientA).

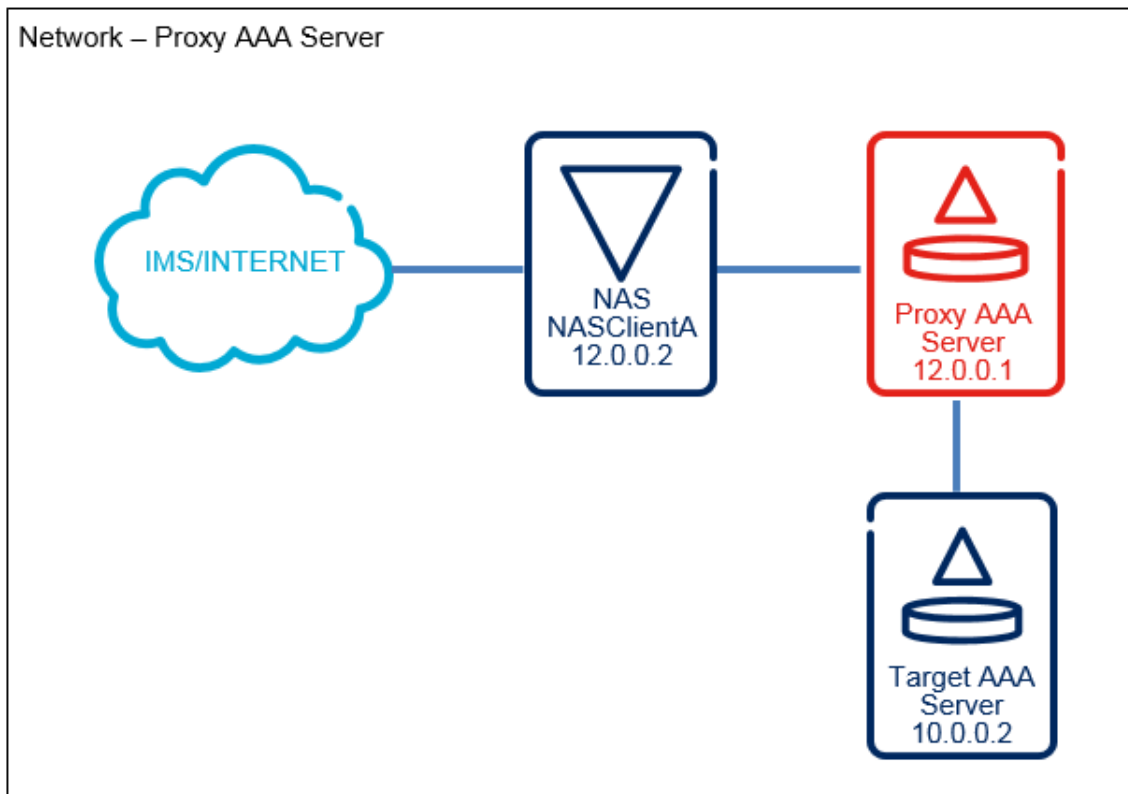


Figure 2 Sample Network 2 - Proxy AAA Server

In Sample Network 2, the proxy AAA server with the IP address of 12.0.0.1 for proxy function. Two types of node are related to the proxy AAA server.

- NAS Client: NASClientA with the IP address of 12.0.0.2 sends and receives Radius messages to and from proxy AAA server. Proxy AAA server will forward Radius request messages to target AAA server and forward Radius response messages to NASClientA.



- Proxy Target: proxy AAA server (10.0.0.2) sends and receives Radius messages to and from target AAA server.

The configuration of proxy AAA server in standalone proxy server includes the following topics:

- Section 4.1 Configuring AAA Server Type on page 28
- Section 4.2 Configuring Proxy Server Network Relationship on page 28
- Section 4.3 Configuring AAA Proxy Function on page 30

4.1 Configuring AAA Server Type

Set the attribute `proxyServerType` in the MO `ProxyControl`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>configure

(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,ProxyControl=1

(config-ProxyControl=1)>proxyServerType=STANDALONE_PROXY_SERVER

(config- ProxyControl=1)>commit

Note: The configuration takes effect after Radius AAA server restarts.

(ProxyControl=1)>exit
```

4.2 Configuring Proxy Server Network Relationship

To configure network relationship between Proxy AAA server and the relevant clients, perform the following procedures:

- Section 4.2.1 Creating Proxy AAA Server on page 28
- Section 4.2.2 Configuring Clients for Proxy AAA Server on page 29
- Section 4.2.3 Configuring Proxy Target for Proxy AAA Server on page 30

4.2.1 Creating Proxy AAA Server

In Sample Network 2, configure a standalone proxy AAA server for standalone proxy function.

1. Create AAA server objects.



Command Syntax:

```
IPWorks> create AAAServer -set name=<AAA Server Name>;address
=<IP Address>
```

For example:

```
IPWorks> create AAAServer -set name=aaasrv1;address=169.254.10
0.3
```

1 object(s) created.

```
IPWorks> create AAAServer -set name=aaasrv2;address=169.254.10
0.4
```

1 object(s) created.

2. Display the created AAA server objects.

Command Syntax:

```
IPWorks> list AAAserver <AAA Server Name>
```

For example:

```
IPWorks> list AAAserver aaasrv1
```

```
[AAAServer aaasrv1]
  Name: aaasrv1
  Address: 169.254.100.3
```

```
IPWorks> list AAAserver aaasrv2
```

```
[AAAServer aaasrv2]
  Name: aaasrv2
  Address: 169.254.100.4
```

4.2.2 Configuring Clients for Proxy AAA Server

In the Sample Network 2, the configured proxy AAA server has one client with the IP address of 12.0.0.2.

Shared secret is configured for the secure communication between the NAS Client and proxy AAA server.

Both IPv4 and IPv6 are supported to work with CUDB.

1. Create an object instance of `ClientSharedSecret` in the MO `ClientSharedSecretMgr`.



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusStack=1,SharedSecretMgr=1,Cli
(config-ClientSharedSecretMgr=1)> ClientSharedSecret=1
```

2. Set the IP address of the client.

```
(config-ClientSharedSecret=1)> clientIPAddr="12.0.0.2"
```

3. Configure the shared secret of the client.

```
(config-ClientSharedSecret=1)> sharedSecretValu
e="AAA-Sharedsecret"
(config-ClientSharedSecret=1)> type=ALL
(config-ClientSharedSecret=1)> commit
(ClientSharedSecret=1)>exit
```

Note: The configuration takes effect immediately.

4.2.3 Configuring Proxy Target for Proxy AAA Server

In Sample Network 2, the configured proxy AAA server has one proxy target with the IP address of 10.0.0.2.

Shared secret is configured for the secure communication between the NAS Client and proxy AAA server.

Both IPv4 and IPv6 are supported to work with CUDB.

1. Create an object instance of `ProxyTargetSharedSecret` in the MO `ProxyTargetSharedSecretMgr`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusStack=1,SharedSecretMgr=1,Pro
(config-ProxyTargetSharedSecretMgr=1)> ProxyT
argetSharedSecret=1
```

2. Set the IP address of the proxy target.

```
(config-ProxyTargetSharedSecret=1)> proxyTarg
etIPAddr="10.0.0.2"
```

3. Configure the Shared secret of the proxy target.

```
(config-ProxyTargetSharedSecret=1)> sharedSecretVa
lue="AAA-Sharedsecret"
(config-ProxyTargetSharedSecret=1)> type=ALL
(config-ClientSharedSecret=1)> commit
(ClientSharedSecret=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

4.3 Configuring AAA Proxy Function

To configure proxy relevant function for AAA, perform the following procedures:



- Section 4.3.1 Configuring Proxy Rule on page 31
- Section 4.3.2 Configuring Proxy Realm on page 31
- Section 4.3.3 Linking Proxy Realm and Proxy Rule on page 33
- Section 4.3.5 Configuring Number of Ports for Each Proxy Server on page 34

4.3.1 Configuring Proxy Rule

1. Create AAA proxy rule.

```
IPWorks> create AAAProxyRule -set name=proxyrule1;req
uestchecklist="User-Name?1";=>
requestchangelist="add:Framed-Protocol=1", "replace:NAS-Port=8000:9000";=>
replychecklist="User-Name?1";replychangelist="add:
Reply-Message='Proxy is OK!'"
```

1 object(s) created.

2. Display information of the created AAA proxy rule.

```
IPWorks> list AAAProxyRule
[AAAProxyRule proxyrule1]
Name: proxyrule1
RequestChecklist: User-Name ? 1
ReplyChecklist: User-Name ? 1
RequestChangelist: add:Framed-Protocol="1", replace:NAS-Port="'8000':'9000'"
ReplyChangelist: add:Reply-Message="Proxy is OK!"
```

4.3.2 Configuring Proxy Realm

Proxy realm applies the configured proxy rule created in Section 4.3.1 Configuring Proxy Rule on page 31 and it can be set to either of the following:

- Section 4.3.2.1 Creating Proxy Realm with Exact Name on page 32
- Section 4.3.2.2 Creating Proxy Realm with Regular Expression on page 32

Prerequisites:

Determine the realm information is got from User-Name, Called-Station-Id, or NAS-Identifier AVP by AAA server. This is configured by the attribute `getRealmFrom` in MO `ProxyControl`.

For example:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IP
(config-ProxyControl=1)>
(config-ProxyControl=1)>getRealmFrom=User-Name
(config-ProxyControl=1)>commit
(ProxyControl=1)>exit
```



Note: The configuration takes effect after Radius AAA server restarts.

The example shows that the AAA server will get the realm information from the AVP User-Name.

4.3.2.1 Creating Proxy Realm with Exact Name

AAA proxy forwards the authentication messages to authdest and forwards the accounting messages to acctdest when the realm name in the message Access-Request/Accounting-Request matches with the proxy realm configured through CLI as follows. If AAA server gets the realm information from User-Name, and if striprealm is set to true, the realm of attribute User-Name is stripped.

For example,

1. Create an AAARealm Object with exact name.

```
IPWorks> create AAARealm Ericsson.com -set striprealm=true;authdest=10.0.0.1;acctdest=10.0.0.1
```

```
1 object(s) created.
```

2. Display the information of the created object.

```
IPWorks> list AAARealm
```

```
[AAARealm Ericsson.com]
Name: Ericsson.com
StripRealm: true
AuthDest: 10.0.0.1
AcctDest: 10.0.0.1
```

4.3.2.2 Creating Proxy Realm with Regular Expression

AAA proxy forwards the authentication messages to authdest and forwards the accounting messages to acctdest when the realm name in the message Access-Request/Accounting-Request matches with the regular expression configured through CLI as follows.

For example,

1. Create an AAARealm Object with regular expression.

```
IPWorks> create AAARealm -set name="/REGEX/^[A-Za-f0-9]{2:}{5}[A-Za-f0-9]{2}$";authdest="10.0.0.1";acctdest="10.0.0.1"
```

2. Display the information of the created object.

```
IPWorks> list aaarealm
```

```
[AAARealm /regex/^[a-fa-f0-9]{2:}{5}[a-fa-f0-9]{2}$]
```



```
Name: /REGEX/^([A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}$
AuthDest: 10.0.0.1
AcctDest: 10.0.0.1
```

3. If you want to specify the exact `aaarealm` with regular expression, take the following as examples to list, modify, delete and select:

```
IPWorks> list aaarealm -where Name="/REGEX/^([A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}$"
```

```
IPWorks> modify aaarealm -where Name="/REGEX/^([A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}$"
```

```
IPWorks> delete aaarealm -where Name="/REGEX/^([A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}$"
```

```
IPWorks> list aaarealm -where Name="/REGEX/^([A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2}$"
```

In this case realm name like : "00:25:b3:e3:3e:d0" is matched with the proxy realm.

4.3.3

Linking Proxy Realm and Proxy Rule

1. Modify proxy realm to link to proxy rule.

```
IPWorks> modify AAAR realm Ericsson.com -set ProxyRule=proxyrule1
```

```
[AAAR realm Ericsson.com]
Name: Ericsson.com
StripRealm: true
AuthDest: 10.0.0.1
AcctDest: 10.0.0.1
```

2. Display the information of the modified object.

```
IPWorks> list AAAR realm Ericsson.com
```

```
[AAAR realm Ericsson.com]
Name: Ericsson.com
StripRealm: true
AuthDest: 10.0.0.1
AcctDest: 10.0.0.1
ProxyRule: proxyrule1
```

4.3.4

Configure the IP and IP Type

To configure the IP and IP type that AAA used for sending/receiving message to/from proxy target, do the following:



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusStack=1,RadiusInterface=1
(RadiusInterface=1)>configure
(config-RadiusInterface=1)> proxyAddress="0.0.0.0"
(config-RadiusInterface=1)> proxyBindIPType=IPv4
(config-RadiusInterface=1)> commit
```

Note: The configuration takes effect after Radius AAA server restarts.

4.3.5 Configuring Number of Ports for Each Proxy Server

It is recommended to adjust the number of ports for each proxy server according to the following formula:

Total Number of Ports = QPS / 255

For example, QPS of proxy function is 25500, then number of ports for each proxy server is set to 100.

1. Configure the attribute proxyPortsNumEachPL in the MO RadiusInterface.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusStack=1,RadiusInterface=1
(RadiusInterface=1)>configure
(config-RadiusInterface=1)>proxyPortsNumEachPL=<Number of Ports>
(config-RadiusInterface=1)>commit
(RadiusInterface=1)>exit
```

2. Restart Radius Stack.

```
# ipw-ctr restart aaa_radius_stack <PL hostname>
```



5 Configuring Home and Proxy AAA Server

Figure 3 shows a sample network for AAA server that works as both home and proxy server.

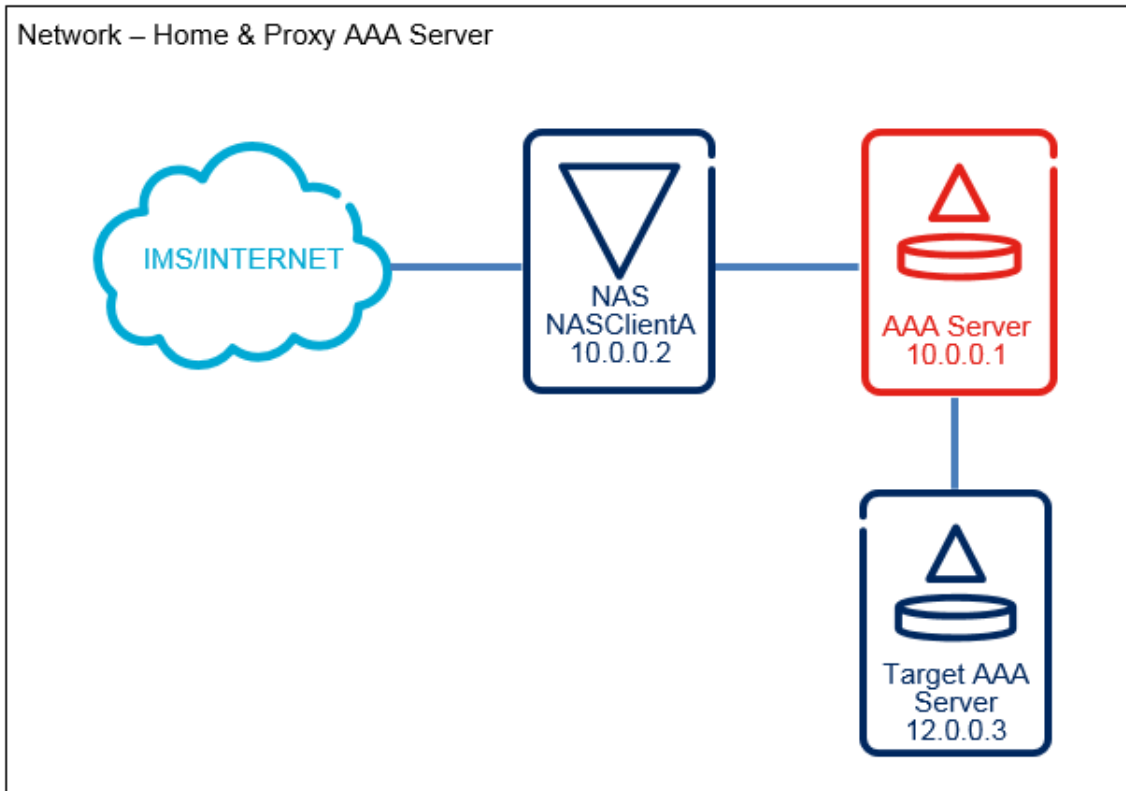


Figure 3 Sample Network 3 - Home & Proxy AAA Server

In Sample Network 3, the proxy functions must be added based on the home AAA server in Sample Network 1. Thus AAA server characteristics will be updated as follows:

- Sample Network 3 includes a AAA (10.0.0.1) server and one NAS Client. Target AAA server (12.0.0.3) plays as a proxy target of the AAA server.
- AAA server with the IP address of 10.0.0.1, which is used to implement Authentication/Authorization/Accounting/Dynamic Authorization and Proxy Function. Two types of nodes are configured for the AAA server :
 - One NAS client: NASClientA with the IP address of 10.0.0.2 sends or receives RADIUS messages to or from the AAA server.
 - One proxy target: The AAA server proxy target with the IP address of 12.0.0.3 receives Radius request messages from or sends Radius response messages to the AAA server (10.0.0.1).



To configure home & proxy AAA server (10.0.0.1), in addition to Section 3 on page 5, proxy target, shared secret, proxy realm, and proxy rule need to be configured.

Both IPv4 and IPv6 are supported to work with CUDB.

1. Configure server type to home and proxy AAA server.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,ProxyControl=1
(config-ProxyControl=1)>proxyServerType=BOTH_HOME_AND_PROXY_SERVER
(config-ProxyControl=1)> commit
(ProxyControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

2. Create an object instance of `ProxyTargetSharedSecret` in the MO `ProxyTargetSharedSecretMgr`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>configure
(config)>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusStack=1,SharedSecretMgr=1,Pro
(config-ProxyTargetSharedSecretMgr=1)> ProxyT
argetSharedSecret=1
```

3. Set the IP address of the client.

```
(config-ProxyTargetSharedSecret=1)> proxyTarg
etIPAddr="12.0.0.3"
```

4. Configure the Shared secret of the client.

```
(config-ProxyTargetSharedSecret=1)> sharedSecretVa
lue="AAA-Sharedsecret"
(config-ProxyTargetSharedSecret=1)> type=ALL
(config-ClientSharedSecret=1)> commit
(ClientSharedSecret=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

5. Configure AAA proxy function, see Section 4.3 Configuring AAA Proxy Function on page 30.



6 Configuring IP Allocation

This section guides how to configure the IPv4 and IPv6 prefix allocation function.

— Prerequisites

To enable AAA IP allocation function, the following conditions must be satisfied:

- Session-based accounting is activated, see Section 3.5.3 Session-based Accounting Configuration on page 16.
- (IPv4 allocation): The AAA session records must include the "User-Name", "Framed-IP-Adress", "Framed-IP-Netmask", and "IP-Alloc-Pool" attributes. See Section 6.1 Configuring AAA Session Records on page 37 for the IPv4 example.
- (IPv6 prefix allocation): The AAA session records must include the "User-Name", "Framed-Ipv6-Prefix", and "pv6-Prefix-Pool" attributes. See Section 6.1 Configuring AAA Session Records on page 37 for the IPv6 prefix example.

— Section 6.2 Configuring IP Allocation Function on page 38

— Section 6.3 Configuring AAA User on page 40

Note: The procedure can be skipped in AAA FE (Radius) scenario.

— Section 6.4 Viewing AAA IP Allocation Status on page 42

— (Optional:) Section 6.5 Force Releasing IPv4 Address and IPv6 Prefix on page 44

6.1 Configuring AAA Session Records

For example:

1. Show information of records in the MO `AAASessionControl`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AAASessionControl=1
(AAASessionControl=1)>show -v
AAASessionControl=1
aaaSessionControlId="1"
acctOnOffCloseAll=true <default>
createProxySession=true <default>
expireDMNotify=false <default>
expireTime=1440 <default>
multiSessionClassPrior=false <default>
records="User-Name,NAS-Port,Framed-IP-Address,Called-Station-Id,⇒
Calling-Station-Id,Acct-Multi-Session-Id" <default>
terminateExpireSession=true <default>
threshold4SessionCapacity="80,90" <default>
updateIPViaAcct=false <default>
```



The example shows that:

- For IPv4 allocation, the `records` attribute does not include the required "Framed-IP-Netmask" and "IP-Alloc-Pool" attributes. Add the attributes by following Step 2.
 - For IPv6 prefix allocation, `records` attribute does not include the required "Framed-Ipv6-Prefix" and "Ipv6-Prefix-Pool" attributes. Add the attributes by following Step 3.
2. Add "Framed-IP-Netmask" and "IP-Alloc-Pool" attributes to the `records` attribute.

```
(AAASessionControl=1)>configure
(config-AAASessionControl=1)> records="User-Name,NAS-Port,Framed-IP-Address
Framed-IP-NetmaskIP-Alloc-Pool,Called-Station-Id,Calling-Station-Id,⇒
Acct-Multi-Session-Id"
(config-AAASessionControl=1)>commit
(AAASessionControl=1)>exit
```

3. Add "Framed-IP-Netmask" and "IP-Alloc-Pool" attributes to the `records` attribute.

```
(AAASessionControl=1)>configure
(config-AAASessionControl=1)> records="User-Name,NAS-Port,Framed-IP-Address
Framed-Ipv6-Prefix,Ipv6-Prefix-Pool,Called-Station-Id,Calling-Station-Id,⇒
Acct-Multi-Session-Id"
(config-AAASessionControl=1)>commit
(AAASessionControl=1)>exit
```

Note: The configuration takes effect after Radius AAA server restarts.

6.2 Configuring IP Allocation Function

This section provides examples of configuring IPv4 and IPv6 prefix allocation function.

Note: IP Allocation Function is not available if **Geography Redundancy** function is enabled.

6.2.1 Configuring IPv4 Allocation Function

To configure IPv4 allocation function, do the following:

1. Create an AAA subnet that contains the IP pool.

```
IPWorks> create aaasubnet subnet1 -set address=
10.0.0.0;masklength=8
```

Note: It is forbidden to modify the subnet and range of created pool. In case of more (or less) IP addresses needed, you can choose to add a new pool, or delete and recreate the pool with different subnet or range.



2. Create an AAA IP pool that contains the IP address that can be allocated.

Note: "ClientIP" and "ClientIdentifier" are alternatives, you can configure one of them, or both. Therefore, either of the following configuration is acceptable.

```
IPWorks> create aaaippool pool1 -set subnet=subnet1;⇒
range=10.0.0.1-100;clientip=10.170.15.41
IPWorks> create aaaippool pool1 -set subnet=subnet1;⇒
range=10.0.0.1-100;clientidentifier=nasclient
IPWorks> create aaaippool pool1 -set subnet=subnet1;⇒
range=10.0.0.1-100;clientip=10.170.15.41;clien
tidentifier=nasclient
```

3. Enable the AAA IP allocation function.

```
>dn ManagedElement=<Node Name>,IpworksFunction=
1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,IPAllocationService=1
(IPAllocationService=1)>configure
(config-IPAllocationService=1)>ipv4alloc=true
(config-IPAllocationService=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts

4. (If needed), delete an AAA IP pool.

Prerequisites: Before delete an AAA IP pool, ensure that not any IP addresses are allocated by using the following command:

```
IPWorks> send aaaserver -message=show -ippool=<poolname>
```

Note: If certain address has been allocated, wait until the IP address is free or force releasing the active IPv4 address of this pool. However, force-release destroys the related session. For details on how to force releasing IPv4 address, see Section 6.5 Force Releasing IPv4 Address and IPv6 Prefix on page 44.

For example, delete an AAA IP pool such as pool11:

```
IPWorks> delete aaaippool pool11
```

Note: The configuration change takes effect after Radius AAA server restart. For more information, see Section 11.1 Restarting Radius AAA Server on page 71.

6.2.2 Configuring IPv6 Prefix Allocation Function

To configure IPv6 prefix allocation function, do the following:

1. Create an AAA IPv6 prefix pool that contains the IPv6 prefix that can be allocated.

Note: "ClientIP" and "ClientIdentifier" are alternatives, you can configure one of them, or both. Therefore, either of the following configuration is acceptable.



```
IPWorks> create aaaipv6prefixpool pool1 -set =>
PrefixRange="2012:ABCD:170::/64-2012:ABCD:170: FFFF::/64";=>
clientip=80fe::226:55ff:fe4c:f58c;
IPWorks> create aaaipv6prefixpool pool1 -set =>
PrefixRange="2012:ABCD:170::/64-2012:ABCD:170: FFFF::/64";=>
clientidentifier=nasclient
IPWorks> create aaaipv6prefixpool pool1 -set =>
PrefixRange="2012:ABCD:170::/64-2012:ABCD:170: FFFF::/64";=>
clientip=80fe::226:55ff:fe4c:f58c;clientidentifier=nasclient
```

Note: It is forbidden to modify the range of created pool. In case of more (or less) IP addresses needed, you can choose to add a new pool, or delete and recreate the pool with different range.

2. Enable the AAA IPv6 prefix allocation function.

```
>dn ManagedElement=<Node Name>,IpworksFunction=
1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,IPAllocationService=1
(IPAllocationService=1)>configure
(config-IPAllocationService=1)>ipv6PrefixAlloc=true
(config-IPAllocationService=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

3. (If needed), delete an AAA IPv6 pool.

Prerequisite: Before delete an AAA IPv6 pool, ensure that not any IP addresses are allocated.

```
IPWorks> send aaaserver -message=show -ipv6prefixpool=<pool_name>
```

Note: If certain address is allocated, wait until the IP address is free or force releasing the active IPv6 address of this pool. However, force-release destroys the related session. For details on how to force releasing IPv6 address, see Section 6.5 Force Releasing IPv4 Address and IPv6 Prefix on page 44.

For example, delete the AAA IPv6 pool such as pool1:

```
IPWorks> delete aaaipv6prefixpool pool1
```

Note: The configuration change takes effect after Radius AAA server restart. For more information, see Section 11.1 Restarting Radius AAA Server on page 71.

6.3 Configuring AAA User

In AAA FE (Radius) scenario, since AAA user configuration data is stored in CUDB, skip this procedure. For more information on the AAA user configuration in CUDB, refer to PG documents.

AAA user can be configured to allocate IPv4 addresses and IPv6 prefix with 4 modes:



Table 1 IPv4 Allocation Mode

Mode	Value	Description	Example
Hint Mode	ipallocotype=0	The IP address is assigned from one or more Radius client related IP address pools if the attribute "Framed-IP-Address" is contained in Access-Request message when the user is authenticated successfully. If the attribute "Framed-IP-Address" is not contained in the Access-Request message, the IP address is not be assigned.	<pre>IPWorks> create aaaippool pool1 -set subnet=subnet1;range=10.0.0.1-100;clientip=10.170.15.41</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipallocotype=0</pre> <p>In this example, Access-Request contains "Framed-IP-Address" and "NAS-IP-Address" attributes. The address of "NAS-IP-Address" is 10.170.15.41. The client is associated with AAAIPPool pool1 that contains the value of "Framed-IP-Address", so the IP address in "Framed-IP-Address" are assigned to the user.</p>
Static Mode	ipallocotype=1	The static IP address is assigned to the user each time when the user is authenticated successfully. The address must not be included in any IP pool that is created by user. The attribute "IPAllocValue" should be set using a static IP address.	<pre>IPWorks> create aaauser user1 -set password=123456;ipallocotype=1;ipallocvalue=10.170.15.123</pre> <p>In this example, 10.170.15.123 is assigned.</p>
Pool Mode	ipallocotype=2	The IP address is assigned from a specific IP address pool. The attribute "IPAllocValue" should be set using an existing AAAIPPool. One available IP address from this specific IP pool is assigned when the user is authenticated successfully.	<pre>IPWorks> create aaaippool pool1 -set subnet=subnet1;range=10.0.0.1-100;clientip=10.170.15.41</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipallocotype=2;ipallocvalue=pool1</pre> <p>If it has the free IP address in the pool, the next free IP in the pool is assigned.</p>
NAS Mode	ipallocotype=3	The IP address is assigned from one or more Radius client related IP address pools. The attribute "IPAllocValue" does not need any value (even if it is configured, AAA server will not handle it). An address is assigned from one of the pools associated with the Radius client when a user is authenticated successfully.	<pre>IPWorks> create aaaippool pool1 -set subnet=subnet1;range=10.0.0.1-100;clientip=10.170.15.41;clientidentify=862133545678</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipallocotype=3</pre> <p>If NAS-IP-Address=10.170.15.41 and NAS-Identify=862133545678, the client is associated with AAAIPPool pool1; If it has free IP address in the pool, the next free IP in the pool is assigned.</p>



Table 2 IPv6 Prefix Allocation Mode

Mode	Value	Description	Example
Hint Mode	ipv6prefixalloctype=0	The IPv6 prefix is assigned from one or more Radius client related IPv6 prefix pools if the attribute "Framed-IPv6-Prefix" is contained in Access-Request message when the user is authenticated successfully. If the attribute "Framed-IPv6-Prefix" is not contained in the Access-Request message, the IPv6 prefix is not be assigned.	<pre>IPWorks> create aaaipv6prefixpool pool1 -set PrefixRange="2012:ABCD:170::/64-2012:ABCD:170:FFFF::/64";clientip=80fe::226:55ff:fe4c:f58c</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipv6prefixalloctype=0</pre> <p>In this example, Access-Request contains "Framed-IP-Address" and "NAS-IPv6-Address" attributes. The value of "NAS-IPv6-Address" is 80fe::226:55ff:fe4c:f58c. The client is associated with the AAAIPv6PrefixPool pool1 that contains the value of "Framed-IPv6-Prefix", so the IP address in "Framed-IPv6-Prefix" is assigned to user.</p>
Static Mode	ipv6prefixalloctype=1	The static IPv6 prefix is assigned to the user each time when the user is authenticated successfully. The IPv6 prefix must not be included in any IPv6 prefix that is created by user. The attribute IPv6PrefixAllocValue should be set using a static IPv6 prefix.	<pre>IPWorks> create aaauser user1 -set password=123456;ipv6prefixalloctype=1;ipv6prefixallocvalue=2012:ABCD:170:29::/64</pre> <p>2012:ABCD:170:29::/64 is assigned.</p>
Pool Mode	ipv6prefixalloctype=2	The IPv6 prefix is assigned from a specific IPv6 prefix pool. The attribute IPv6PrefixAllocValue should be set using an existent AAAIPv6PrefixPool. One available IPv6 prefix from this specific IPv6 prefix pool will be assigned when the user is authenticated successfully.	<pre>IPWorks> create aaaipv6prefixpool pool1 -set PrefixRange="2012:ABCD:170::/64-2012:ABCD:170:FFFF::/64";clientip=80fe::226:55ff:fe4c:f58c</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipv6prefixalloctype=2;ipv6prefixallocvalue=pool1</pre> <p>If free IPv6 prefix exists in the pool, the next free IPv6 prefix in the pool will be assigned.</p>
NAS Mode	ipv6prefixalloctype=3	The IPv6 prefix is assigned from one or more Radius client related IPv6 prefix pools. The attribute IPv6PrefixAllocValue does not need any value (even if it is configured, AAA server will not handle it). An IPv6 prefix is assigned from one of the pools associated with the Radius client when a user is authenticated successfully.	<pre>IPWorks> create aaaipv6prefixpool pool1 -set PrefixRange="2012:ABCD:170::/64-2012:ABCD:170:FFFF::/64";clientip=80fe::226:55ff:fe4c:f58c;clientidentify=862133545678</pre> <pre>IPWorks> create aaauser user1 -set password=123456;ipv6prefixalloctype=3</pre> <p>If NAS-IPv6-Address=80fe::226:55ff:fe4c:f58c and NAS-Identify=862133545678, the client is associated with aaaipv6prefixpool pool1; If free IPv6 prefix exists in the pool, the next free IPv6 prefix in the pool will be assigned.</p>

6.4 Viewing AAA IP Allocation Status

The user can view the status of AAA IPv4 or IPv6 prefix pool using the send command:

IPv4:



For example:

```
IPWorks> send aaaserver -message=show -ippool=pool1
```

```
[AAAIPPool pool1]
  Name: pool1
  Subnet: subnet1
  AddressRange: 10.0.0.1-100
  Total IP Address Count: 100
  Allocated IP Address Count: 0
  Allocated IP Address Percentage: 0%
```

IPv6 prefix:

For example:

```
IPWorks> send aaaserver -message=show -ipv6prefixpool=pool1
```

```
[AAAIIPv6PrefixPool pool1]
  Name: pool1
  PrefixRange: 2012:ABCD:170::/64-2012:ABCD:170:FFFF::/64
  Total IPv6 Prefix Count: 65535
  Allocated IPv6 Prefix Count: 0
  Allocated IPv6 Prefix Percentage: 0%
```

The user can also view the status of AAA session that is related to the IPv4 pool and IPv4 address, or the IPv6 pool and IPv6 prefix:

IPv4:

```
IPWorks> list aaasession -where ipallocpool=pool1
```

Or

```
IPWorks> list aaasession -where framedipaddress=10.0.0.1
```

```
[AAASession aaa1]
  UniqueSessionId: sess21276727519
  NasId: 466
  NasType: nas-identifier
  AcctSessionId: 1310
  StartTime: 2010:06:17:06:31:59
  Status: init
  UserName: user1
  FramedIpAddress: 10.0.0.1
  IpAllocPool: pool1
  LastUpdateTime: 2010:06:17:06:31:59
```

IPv6 prefix:



```
IPWorks> list aaasession -where ipv6prefixpool=pool1
```

Or

```
IPWorks> list aaasession -where framedipv6prefix=2012:ABCD:170:29::/64
```

```
[AAASession aaa1]
UniqueSessionId: sess21276727519
NasId: 466
NasType: nas-identifier
AcctSessionId: 1310
StartTime: 2010:06:17:06:31:59
Status: init
UserName: user1
FramedIpv6Prefix: 2012:ABCD:170:29::/64
IPv6 Prefix Pool: pool1
LastUpdateTime: 2010:06:17:06:31:59
```

6.5 Force Releasing IPv4 Address and IPv6 Prefix

The user is able to release the active IPv4 address leases of AAA IPv4 address pool.

For example:

```
IPWorks> send aaaserver -message=forcerelease -ippool=pool1
```

Note: The force-release destroys the session in which IPs are allocated from the pool. This means, if an IPv6 prefix is associated with the same session, the IPv6 prefix is released as well.

The user is able to release the active IPv6 prefix lease of AAA IPv6 prefix pool.

For example:

```
IPWorks> send aaaserver -message=forcerelease -ipv6prefixpool=pool1
```

Note: If an IPv4 address is associated with the same session, the IPv4 address is released as well.



7 Configuring AAA Front End (Radius)

In this section, it describes how to configure AAA Front End (Radius) by following topics:

- Section 7.1 Enabling AAA Front End (Radius) Feature on page 45
- Section 7.2 Configuring CUDB Connection Pool on page 45
- Section 7.3 Configuring LDAP Dictionary on page 48
- Section 7.4 Configuring AAA Front End (Radius) Graceful Handling for CUDB Overload Protection on page 49
- Section 7.5 Configuring IPsec Tunnel for CUDB on page 50
- Section 7.6 Configuring AAA Front End (Radius) Counter in CUDB on page 50

7.1 Enabling AAA Front End (Radius) Feature

To enable AAA FE (Radius) feature, do the following command:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node name>,IpworksFunction=1,IPWorksAAARoot=1
,IPWorksRadiusAAARoot=1,RadiusAAAService=1,RadiusAAAFEService=1
(RadiusAAAFEService=1)>configure
(config-RadiusAAAFEService=1)>fEServiceEnable=true
(config-RadiusAAAFEService=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

7.2 Configuring CUDB Connection Pool

This section guides how to configure CUDB connection pool for AAA.

Prerequisite:

If route is required for the CUDB connection, follow the examples described in *Configure Route for IPWorks PL Node*.

Table 3 lists the presupposition values that are used as an example for the CUDB connection configuration. CUDB connection configuration varies based on the actual environment. Both IPv4 and IPv6 are supported to work with CUDB.



Table 3 Example: CUDB Node Parameters Values

CUDB Site Name	CUDB Node Name ⁽¹⁾	CUDB Node Parameters
site1	node1	Address = "192.168.20.11"
		distinguishedName = ""
		Password = ""
		poolSize = 16
		Port = 389
	node2	Address = "192.168.20.12"
		distinguishedName = "cudbUser=AAAUser,ou=admin,dc=ericsson,dc=com"
		Password = "secret"
		poolSize = 16
		Port = 389
site2	node1	Address = "192.168.20.13"
		distinguishedName = ""
		Password = ""
		poolSize = 16
		Port = 389
	node2	Address = "192.168.20.14"
		distinguishedName = ""
		Password = ""
		poolSize = 16
		Port = 389
site3	node1	Address = "192.168.20.15"
		distinguishedName = ""
		Password = ""
		poolSize = 16
		Port = 389
	node2	Address = "192.168.20.16"
		distinguishedName = ""
		Password = ""
		poolSize = 16
		Port = 389

(1) Only CUDB AD Node can be used for connection.

Follow the following example to configure the other CUDB sites and CUDB nodes.



```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IpworksCommonRoot=1,
DataBaseInfo=1,CudbManager=1,CudbServiceSite=AAA,CudbSiteManager=1
(CudbSiteManager=1)>configure
(config-CudbSiteManager=1)>CudbSite=site1
(config-CudbSite=site1)>CudbNode=node1
(config-CudbNode=node1)>address=192.168.20.11
(config-CudbNode=node1)>poolSize=16
(config-CudbNode=node1)>show -v
CudbNode=node1
  address="192.168.20.11"
  cudbNodeId="node1"
  distinguishedName=[] <empty>
  password=[] <empty>
  poolSize=16
  port=389 <default>
(config-CudbNode=node1)>up
(config-CudbSite=site1)>CudbNode=node2
(config-CudbNode=node2)>address=192.168.20.12
(config-CudbNode=node2)>distinguishedName="cudbUser=AAAUser,ou=admin,dc=ericsson,dc=com"
(config-CudbNode=node2)>password="secret" cleartext
(config-CudbNode=node2)>poolSize=16
(config-CudbNode=node2)>show -v
CudbNode=node2
  address="192.168.20.12"
  cudbNodeId="node2"
  distinguishedName="cudbUser=AAAUser,ou=admin,dc=ericsson,dc=com"
  password="1:k8d2jPCL2Qa76V1jmjN6+CLUQIbQreeg"
  poolSize=16
  port=389 <default>
(config-CudbNode=node2)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

- <Node Name> represents the node name for IPWorks.
- The username and password of the cudbUser are created by CUDb. If they are used, make sure that they have been created in CUDb before you restart Radius AAA server. If not, the related configuration is unnecessary and keeps it empty.

The final results are as below:

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IpworksCommonRoot=1,
DataBaseInfo=1,CudbManager=1,CudbServiceSite=AAA,CudbSiteManager=1
(CudbSiteManager=1)>show -v CudbSite=site1
CudbSite=site1
  cudbSiteId="site1"
  CudbNode=node1
  CudbNode=node2
(CudbSiteManager=1)>show -v CudbSite=site2
CudbSite=site2
  cudbSiteId="site2"
  CudbNode=node1
  CudbNode=node2
(CudbSiteManager=1)>show -v CudbSite=site3
CudbSite=site3
  cudbSiteId="site3"
  CudbNode=node1
  CudbNode=node2
```

Note: The total pool size is recommended to be smaller than 1000 according to IPWorks environment and CUDb configuration.

$$\begin{aligned} \text{total_pool_size} &= \text{site1_pool_size} + \text{site2_pool_size} + \\ &\text{site3_pool_size} = 200 + 200 + 400 = 800 < 1000 \end{aligned}$$



CUDB Site Priority

In the scenario of CUDB connection pool with multiple sites, for example, three CUDB sites (site1, site2, and site3), the site priority is as below:

site1>site2>site3

Which means:

- AAA-FE (Radius) or NP connects to the nodes in site1 by default.
- If site1 is unreachable, then it connects to site2.
- If both site1 and site2 are unreachable, then it connects to site3.
- If site1 has recovered, then it switches back to site1.

AAA-FE (Radius) does not connect to a lower priority site if a higher priority site (like site1) is available or recovered.

CUDB Site Priority Strategy

The CUDB site with a lower string name has a higher priority.

String name <X> is lower than string name <Y> in the following cases:

- Both string names are compared character by character. The value of the first unmatched character in string name <X> is lower than the character in string name <Y>. For example, site1 > site2.
- All compared characters match but string name <X> is shorter.

7.3 Configuring LDAP Dictionary

```
# vi /etc/ipworks/ldapschema/ldap_dictionary.xml
```

```
...
<service name="AAA">
  <cudbRootEntry name="dc=o,dc=com"/>
  <bindDn name="ou=identities,"/>
  <searchDn name="serv=AA,UserName=%s,dc=UserName,ou=identities,"/>
  <entryList>
    <entry name="profile">
      <dn name="serv=AA,mscId=%s,ou=multiSCs,"/>
      <attr name="UserName" alias="UserName"/>
      <attr name="userPassword" alias="Password"/>
      <attr name="AuthMethod" alias="AuthMethod"/>
      <attr name="IPAllocType" alias="IPAllocType"/>
      <attr name="IPAllocValue" alias="IPAllocValue"/>
      <attr name="IPv6PrefixAllocType" alias="IPv6PrefixAllocType"/>
      <attr name="IPv6PrefixAllocValue" alias="IPv6PrefixAllocValue"/>
    </entry>
  </entryList>
</service>
```



```

        <attr name="GroupNameList" alias="GroupNameList"/>
        <attr name="PolicyNameList" alias="PolicyNameList"/>
    </entry>
    <entry name="individualPolicy">
        <dn name="PolicyName=%s,serv=AA,mscId=%s,ou=multiSCs,"/>
        <attr name="PolicyName" alias="PolicyName"/>
        <attr name="PolicyChecklist" alias="PolicyChecklist"/>
        <attr name="PolicyReplylist" alias="PolicyReplylist"/>
    </entry>
    <entry name="group">
        <dn name="GroupName=%s,ou=Groups,serv=AA,ou=mscCommonData,"/>
        <attr name="GroupName" alias="GroupName"/>
        <attr name="PolicyNameList" alias="PolicyNameList"/>
    </entry>
    <entry name="sharePolicy">
        <dn name="PolicyName=%s,ou=Policies,serv=AA,ou=mscCommonData,"/>
        <attr name="PolicyName" alias="PolicyName"/>
        <attr name="PolicyChecklist" alias="PolicyChecklist"/>
        <attr name="PolicyReplylist" alias="PolicyReplylist"/>
    </entry>
</entryList>
...

```

Make sure that the content is aligned with customer LDAP server configuration: cudbRootEntry, searchDn and dn.

7.4 Configuring AAA Front End (Radius) Graceful Handling for CUDB Overload Protection

To configure AAA FE (Radius) graceful handling for CUDB Overload Protection, execute the following command:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IpworksCommonRoot=1,DataBaseInfo=1,CudbManager=1,CudbFunction=1
```

Parameter	Default Value	Description
maxRejectRate	95	The Max rate of rejecting or discarding Access-Request in CUDB overload protection situation.
busyRateThreshold	2	<p>The threshold value in percentage of busy response numbers (from CUDB node) and ldap requests number (from IPWorks). When the real value exceeds this threshold value, IPWorks starts to discard Access-Request.</p> <p>The threshold value in percentage of the rate: number of LDAP_BUSY(received from CUDB) / number of queries (sent to CUDB).</p>



Parameter	Default Value	Description
rejectRateUpStep	5	The step value in percentage used in recovery procedure from CUDB overload protection. The next rejection rate is the previous rejection rate minus the step value.
rejectRateDownStep	10	The step value in percentage used in continuous CUDB overload protection situation. The next rejection rate is the step value plus the previous rejection rate.

Note:

- Do not change the value of the other parameters. If the default value does not meet the requirements of user, contact the site engineer for support.
- The configuration takes effect after Radius AAA server restarts.

7.5 Configuring IPsec Tunnel for CUDB

If the operator wants to apply IPsec function, refer to [eVIP Management Guide](#) for more information.

7.6 Configuring AAA Front End (Radius) Counter in CUDB

If needed, configure the counter AAAUSERCNT in CUDB.

For more information about the counter and how to do the configuration, refer to [IPWorks Application Counters in CUDB](#).



8 Configuring Wi-Fi AAA

This section describes the procedures of configuring Wi-Fi AAA. IPWorks supports two EAP authentication method: EAP-SIM, EAP-AKA.

Prerequisites

Before the configuration, the following prerequisite must be met:

- SS7 Stack must be configured for WiFi AAA if the user case is 2G/3G USIM (SIM) based on UE authentication with HLR. For more information, refer to [Configure SS7 for AAA](#).

To configure Wi-Fi AAA, perform the following procedures:

1. Section 3.1 Configuring AAA Server Type on page 5.
2. Section 3.2 Configuring Home AAA Server Network Relationship on page 6.
3. Section 3.3 Configuring Radius Stack Interface on page 8.
4. Section 3.4.1 Configuring Authentication and Authorization Selector on page 8.
5. Section 3.5.1 Configuring CSV Record on page 14 and Section 3.5.3 Session-based Accounting Configuration on page 16
6. Section 3.7.1.2 Changing to NAS and Class Correlation Working Mode on page 22, only NAS and Class Correlation Working Mode is supported.
7. Section 8.1 Configuring EAP Method on page 52.
8. Section 8.2 Configuring Local SS7 Parameters on page 53.
9. Section 8.3 Configuring Subscription Authorization Mode on page 53.

According to the your actual requirement, choose to perform the following procedure(s):

- Section 8.4 Configuring CUI Switch on page 58
- Section 8.5 Configuring Showing IMSI in Access-Accept Package on page 58
- Section 8.6 Configuring Subscriber-Charging-Characteristics in Access-Accept on page 59
- Section 8.7 Configuring GT Convert on page 59
- Section 8.8 HSS Integration for Wi-Fi AAA Configuration on page 61



8.1 Configuring EAP Method

Operator can determine which EAP methods should be supported. Select the EAP method for user by the following steps:

1. Enable the EAP method by setting the attribute enable in each EAP method (EAPSIM, EAPAKA, EAPMD5).

If EAPAKA/EAPSIM is enabled, check or modify the `identityFormat` in these two methods, IPWorks AAA selects the EAP method according to EAP identity value format.

2. Configure EAP method selector.

- a. Configure the mode in `EapMethodSelector`.

Based on different configured mode, IPWorks picks up EAP method with different priority.

For example:

Identity first: AAA server uses the EAP-Response/Identity value to match all the defined regular express. If failed, AAA server tries to search the database.

- b. Configure the `defaultEap`.

If the identity format or the database user profile cannot help locate a EAP method, AAA server uses a predefined default EAP method as proposal. Make sure that the configured EAP method is enabled.

To configure EAP method, execute the following command:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

> dn ManagedElement=<Node Name>,IpworksFunction=1,IPWork
sAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,Aut
hMethodControl=1,EapMethodControl=1
(EapMethodControl=1)>show -v --recursive
EapMethodControl=1
  eapMethodControlId="1"
  EapAKA=1
    eapAKAId="1"
    enable=true <default>
    failedNotification=false <default>
    fastReauthMaxCounter=5 <default>
    identityFormat="^[024].+"
    requestVectorNum=5 <default>
  EapMD5=1
    eapMD5Id="1"
    enable=true
    failedNotification=false <default>
```




```

        needAuthorization=true <default>
EapMethodSelector=1
    defaultEap=EAP_METHOD_MD5 <default>
    eapMethodSelectorId="1"
    mode=IDENTITY_ONLY <default>
EapSIM=1
    eapSIMId="1"
    enable=true <default>
    failedNotification=false <default>
    fastReauthMaxCounter=5 <default>
    identityFormat="^[135].+" <default>
    requestVectorNum=5 <default>
    requiredVectorNum=3 <default>

```

Note: The configuration takes effect after Radius AAA server restarts.

8.2 Configuring Local SS7 Parameters

To configure WLAN SS7 SGSN, you must configure the attributes `cpmAddress`, `isdnNumber`, `isdnNumberNature`, `originalSignalingPointCode` and `sgsnAddress` of MO `RadiusSS7Stack`.

For example:

```

# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWork
sAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,I
WLANService=1,RadiusSS7Stack=1
(RadiusSS7Stack=1)>show -v
RadiusSS7Stack=1
    cpmAddress="ss7cafcpmaddress:6669"
    isdnNumber="123456"
    isdnNumberNature=NOA_NATIONAL_SIGNIFICANT <default>
    nodeType=1 <default>
    numberOfAAAProcess=10
    numberOfBEInstance=10
    originalSignalingPointCode=100 <default>
    radiusSs7StackId="1"
    sgsnAddress="192.168.20.13"
    useGT4CallingPartyAddress=false <default>

```

Note: The configuration takes effect after Radius AAA server restarts.

8.3 Configuring Subscription Authorization Mode

The authorization mode can be `APN_MODE` or `ODB_MODE`. You can see the specific configuring procedure in this section.



For more information, refer to IPWorks Wi-Fi AAA Function Overview.

8.3.1 Configuring Subscription Authorization in APN_Mode

In APN_Mode, configure the `subscriptionAuthzMode` of MO `IWLANConfig` to `APN_MODE`.

Then, apply one of the following configuration procedure according to different feature enabled.

- Configuring APN with Wi-Fi Subscription
- Configuring APN Mode without Wi-Fi Subscription

8.3.1.1 Configuring APN with Wi-Fi Subscription

Without Supporting S2a Scenario:

To configure APN without supporting S2a scenario, do the following:

1. Disable the S2a scenario support feature by configuring the parameter `S2aEnabled` of MO `IWLANConfig` as `true`.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRo
otId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAA
Service=1,IWLANSservice=1,IWLANSconfig=1
(IWLANSconfig=1)>configure
(config-IWLANSconfig=1)>s2aEnabled=false
(config-IWLANSconfig=1)>commit
```

2. Configure the attributes of APN-Prefix.

- a. Open the file.

```
# vi /etc/ipworks/aaa_radius/aaa_wifi_data.xml
```

- b. Modify the attributes of APN-Prefix.

For more information about APN-Prefix, refer to Table 4 and Table 5.

- c. Save the file and exit.

Note: The configuration takes effect after Radius AAA server restarts.

With Supporting S2a Scenario:

To configure APN with supporting S2a scenario, do the following:



1. Enable the S2a scenario Support feature by configuring the parameter S2aEnabled of MO IWLANConfig as true.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>s2aEnabled=true
(config-IWLANConfig=1)>commit
```

Note: If s2aEnabled=true, the Access-Request packet includes the CUI AVP regardless whether mandatoryCUI of MO IWLANConfig is true or false.

2. Configure the attributes of APN-Prefix and S2a-GTP-Tunnel-Parameter.
 - a. Open the file.

```
# vi /etc/ipworks/aaa_radius/aaa_wifi_data.xml
```

- b. Modify the attributes of APN-Prefix and S2a-GTP-Tunnel-Parameter.

For more information about APN-Prefix and S2a-GTP-Tunnel-Parameter, refer to Table 4 and Table 5.

3. Save the file and exit.

Note: The configuration takes effect after Radius AAA server restarts.

8.3.1.2

Configuring APN Mode without Wi-Fi Subscription

Without Supporting S2a Scenario:

To configure APN without supporting S2a scenario, do the following:

1. Disable the S2a Scenario Support feature by configuring the parameter s2aEnabled of MO IWLANConfig as false.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>s2aEnabled=false
(config-IWLANConfig=1)>commit
```



2. Enable the APN without Wi-Fi subscription indication feature by configuring the parameter `gprsApnCheckEnabled` of MO `IWLANConfig` as true.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>gprsApnCheckEnabled=true
(config-IWLANConfig=1)>commit
```

3. Configure the APN-Blacklist by modifying the file `/etc/ipworks/aaa_radius/aaa_wifi_data.xml`.

```
# vi/etc/ipworks/aaa_radius/aaa_wifi_data.xml
```

The value of APN Name should be replaced by the name of operator when configuring file.

For more information about the configuration file, see Section 12.2 Example for Wi-Fi AAA on page 75.

4. Save the changes and exit.

Note: The configuration takes effect after Radius AAA server restarts.

With Supporting S2a Scenario:

To configure APN with supporting S2a scenario, do the following:

1. Enable the S2a scenario feature by configuring the parameter `S2aEnabled` of MO `IWLANConfig` as true.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>s2aEnabled=true
(config-IWLANConfig=1)>commit
```

2. Enable the APN without Wi-Fi subscription indication feature by configuring the parameter `gprsApnCheckEnabled` of MO `IWLANConfig` as true.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
```



```
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>gprsApnCheckEnabled=true
(config-IWLANConfig=1)>commit
```

3. Configure the S2a-APN-List, APN-Blacklist, GTP tunnel by modifying the file `/etc/ipworks/aaa_radius/aaa_wifi_data.xml`.

```
# vi/etc/ipworks/aaa_radius/aaa_wifi_data.xml
```

The value of APN Name and GTP tunnel should be replaced by the name of operator when configuring file.

For more information about the configuration file, see Section 12.2 Example for Wi-Fi AAA on page 75.

4. Save the changes and exit.

Note: The configuration takes effect after Radius AAA server restarts.

8.3.2

Configuring Subscription Authorization in ODB_Mode

You should configure the attributes `subscriptionAuthzMode` and `odbHPLMNDDataBit` of MO `IWLANConfig`.

```
#ssh <username>@<MIP_OAM_IP> -t -s cli
```

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>show -v
IWLANConfig=1
aaaServerId="0" <default>
acctInterimInterval=600 <default>
gprsApnEnabled=false <default>
iWlanConfigId="1"
mandatoryChargingCharacter=false <default>
mandatoryCUI=true
odbHPLMNDDataBit=2 <default>
s2aEnabled=false <default>
sessionTimeout=86400 <default>
subscriptionAuthzMode=ODB_MODE <default>
terminationAction=NO_RADIUS_REQUEST <default>
userNameShowIMSI=false <default>
userRateType=0 <default>
wlan3gppChargingCharacter="0000" <default>
wlan3gppGprsNegQosProfile="05-0A11012901010111050101100101" <default>
```

Note: The configuration takes effect after Radius AAA server restarts.



8.4 Configuring CUI Switch

According to section 2.1 in RFC 4372 ([Chargeable-User-Identity \(CUI\)](#)) should not be replied if it is not contained the Access-Request packet. By default, the AAA server does not include the CUI attribute in the Access-Accept packet if the Access-Request packet does not contain the CUI. However, in some solutions, the NAS client does not send CUI in the Access-Request packet to the AAA server but demands the reply of the CUI in the Access-Accept packet. For example, NetOp PM, which is the NAS client for AAA server, does not send CUI, but returning the MSISDN is a key issue to the solution.

To support these solutions, a switch to control the CUI is developed by introducing the attribute `mandatoryCUI` of MO `IWLANConfig`. When the attribute is set to false (default value), CUI will not be sent from the AAA server; when the attribute is set to true, CUI will be sent from the AAA server.

Follow these substeps to configure this parameter:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>mandatoryCUI=false
(config-IWLANConfig=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

Note: If S2a is enabled (`s2aEnabled=true`), the Access-Request packet includes the CUI AVP regardless whether `mandatoryCUI` is true or false.

8.5 Configuring Showing IMSI in Access-Accept Package

This section guides how to configure AAA to put the user IMSI into User-Name AVP in Access-Accept packet in EAP-AKA/SIM authentication.

To support this function, do the following:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>userNameShowIMSI=false
(config-IWLANConfig=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.



Note: Since it is not recommended to unnecessarily send un-ciphered subscriber data (IMSI) over air interface, the last trusted radius proxy on the way down to the subscriber should remove this information before sending it to the subscriber. In the current solution this will be NetOP PM.

8.6 Configuring Subscriber-Charging-Characteristics in Access-Accept

This section guides the configuration personnel how to configure AAA to send the Subscriber-Charging-Characteristics information to the network elements. By doing so, the post-paid and pre-paid subscribers are differentiated. The "Subscriber-Charging-Characteristics" is mapped into a Radius Vendor Specific AVP and then extend it to NetOP Policy Manager.

Technically, the AAA server retrieves the Subscriber-Charging-Characteristics from HLR; however, if HLR does not send Subscriber-Charging-Characteristics to AAA, the AAA can generate it by itself. To realize this mechanism, the operator must configure a proper value for the attribute `wlan3gppChargingCharacter` of MO `IWLANConfig`.

To support this function, do the following:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction.ipworksRootId=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,IWLANSservice=1,IWLANSConfig=1,IWLANConfig=1
(IWLANConfig=1)>configure
(config-IWLANConfig=1)>wlan3gppChargingCharacter="1234"
(config-IWLANConfig=1)>commit
```

Note: The configuration takes effect after Radius AAA server restarts.

8.7 Configuring GT Convert

Sometimes, customers use the Mobile Global Title (MGT, E.214) to address the HLR. The MGT is a result of IMSI Series Analysis.

IPWorks supports converting the IMSI to E.214 based Global Title (GT) from the GT Convert config file. IPWorks matches the GTConvert configuration item according to the prefix of IMSI.

- If one match is found, then IPWorks encodes E.214 format GT, otherwise still encodes original E.212 format GT.
- If several GTConverts match the IMSI number, the first one selected by the longest prefix match will be applied.

Users are required to create/modify the instance of MO `GTConvert`. Here is an example:



```
# ssh <username>@<MIP_OAM_IP> -t -s cli

>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksAAACCommonRoot=1,GTConvertManager=1
(GTConvertManager =1)>configure
(config-GTConvertManager=1) >
(config-GTConvertManager=1)>GTConvert=1
(config-GTConvert=1)>IMSI Series=46000
(config-GTConvert=1)>numOfDigitToRemove=5
(config-GTConvert=1)>digitsToAdd=86139
(config-GTConvert=1)>natureOfAddress=NOA_NATIONAL_SIGNIFICANT
(config-GTConvert=1)>commit
(GTConvert=1)>show -v
GTConvert=1
    digitsToAdd="86139" <default>
    gtConvertId="1"
    IMSISeries="46000"
    natureOfAddress=NOA_NATIONAL_SIGNIFICANT
    numOfDigitToRemove=5 <default>
(GTConvert=1)>up
(GTConvertManager=1)>configure
(config-GTConvertManager=1)>GTConvert=2
(config-GTConvert=2)>IMSI Series=24000
(config-GTConvert=2)>numOfDigitToRemove=5
(config-GTConvert=2)>digitsToAdd=86137
(config-GTConvert=2)>natureOfAddress=NOA_NETWORK_SPECIFIC
(config-GTConvert=2)>commit
(GTConvert=2)>show -v
GTConvert=2
    digitsToAdd="86137"
    gtConvertId="2"
    IMSISeries="24000"
    natureOfAddress=NOA_NETWORK_SPECIFIC
    numOfDigitToRemove=5 <default>
(GTConvert=2)>up
(GTConvertManager=1)>show -v --recursive
GTConvertManager=1
    gtConvertManagerId="1"
    GTConvert=1
        digitsToAdd="86139" <default>
        gtConvertId="1"
        IMSISeries="46000"
        natureOfAddress=NOA_NATIONAL_SIGNIFICANT
        numOfDigitToRemove=5 <default>
    GTConvert=2
        digitsToAdd="86137"
        gtConvertId="2"
        IMSISeries="24000"
        natureOfAddress=NOA_NETWORK_SPECIFIC
        numOfDigitToRemove=5 <default>
(GTConvertManager=1)>
```




Note: The configuration takes effect after Radius AAA server restarts.

8.8

HSS Integration for Wi-Fi AAA Configuration

This section guides the configuration personnel how to perform the HSS integration for Wi-Fi AAA.

Prerequisites:

Before the configuration, the following prerequisites must be met:

- Wi-Fi AAA function has been configured. For the Wi-Fi AAA configuration procedures, see the list in Section 8 on page 51.
- SS7 Stack must be configured for Wi-Fi AAA if the use case is 2G/3G USIM(SIM) based UE authentication with HLR, and for more information, refer to the section Configuring SS7 for Wi-Fi AAA in [Configure SS7 for AAA](#).
- Radius and EPC servers are running normally.

To realize the HSS integration for Wi-Fi AAA, you need to enable IPWorks to forward the Radius AAA messages to the EPC AAA server:

1. Configure the Translation Agent Mode.

- a. Log on to the ECLI.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

- b. Enter the configuration mode.

```
>configure
```

- c. Set the value of `tranAgentMode` to `HSS_ONLY` or `HSS_PRE`.

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,TrustWIFIService=1
```

```
(config-TrustWIFIService=1)>tranAgentMode=HSS_PRE
```

2. Enable EAP-AKA:

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,RadiusAAAService=1,AuthMethodControl=1,EapMethodControl=1,EapAKA=1
```

```
(config-EapAKA=1)>enable=true
```

```
(config-EapAKA=1)>commit
```



Note:

- Currently, IPWorks only supports EAP-AKA and EAP-AKA' method forwarding.
 - The configuration takes effect after Radius AAA server restarts.
3. For 4G based UE, the configuration for the EAP-AKA behavior is performed on the MO `EapAkaConfig` instead of the MO `EapMethodControl`. For more formation on how to configure the EPA-AKA behavior, refer to the section [Configuring EAP AKA/AKA' Authentication in Configure EPC AAA](#).



9

Configuring Accounting Forward/Mediation

This section describes the configuration of AAA to forward Accounting Request message to one or more groups of remote servers.

Prerequisites

Before configuration, the following prerequisites must be met:

- Session-based accounting is activated, see Section 3.5.3 Session-based Accounting Configuration on page 16. To understand the whole process of session-based accounting, refer to section Accounting Session in IPWorks Generic AAA Function Overview.
- Home & proxy AAA server is configured, see Section 5 on page 35 for the detailed configuration.

To configure the Accounting Forward/Mediation function, do the following:

1. Enable Accounting Forward/Mediation function.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,
IPWorksRadiusAAARoot=1,RadiusAAAService=1,AccountingService=1,AcctForward
(AcctForward=1)>configure
(config-AcctForward=1)>enable=true
(config-AcctForward=1)>commit
(AcctForward=1)>exit
```

2. Configure the shared secret for the remote server.

In the following example, you add two shared secrets for the remote server:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRa
RadiusStack=1,SharedSecretMgr=1,ProxyTargetSharedSecretMgr=1
(ProxyTargetSharedSecretMgr=1)>configure
(config-ProxyTargetSharedSecretMgr=1)>ProxyTargetSharedSecret=1
(config-ProxyTargetSharedSecret=1)>proxyTargetIPAddr=6.132.64.5
(config-ProxyTargetSharedSecret=1)>sharedSecretValue=123456
(config-ProxyTargetSharedSecret=1)>
(config-ProxyTargetSharedSecret=1)>up
(config-ProxyTargetSharedSecretMgr=1)>ProxyTargetSharedSecret=2
(config-ProxyTargetSharedSecret=2)>proxyTargetIPAddr=6.132.64.6
(config-ProxyTargetSharedSecret=2)>sharedSecretValue=123456
(config-ProxyTargetSharedSecret=2)>commit
(ProxyTargetSharedSecret=2)>exit
```

3. Configure the Accounting Forward/Mediation group and associated trigger condition.

In the following example, you add two Accounting Forward/Mediation groups with different trigger conditions:



```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRadiusAAARoot=1,
RadiusAAAService=1,AccountingService=1,AcctForward=1,AcctForwardGroupMgr=1
(AcctForwardGroupMgr=1)>configure
(config-AcctForwardGroupMgr=1)>AcctForwardGroup=2
(config-AcctForwardGroup=2)>triggerAVPs=["NAS-Identifier=/REGEX/10.*", "NAS-IP-Address=*"]
(config-AcctForwardGroup=2)>acctForwardDestAddr="6.132.64.5,6.132.64.6"
(config-AcctForwardGroup=2)>up
(config-AcctForwardGroupMgr=1)
(config-AcctForwardGroupMgr=1)>AcctForwardGroup=3
(config-AcctForwardGroup=3)>acctForwardDestAddr="6.132.64.7,6.132.64.8"
(config-AcctForwardGroup=3)>triggerChecklist=["(NAS-Identifier == 10086) || (NAS-IP-Address =
(config-AcctForwardGroup=3)>commit
(AcctForwardGroup=3)>exit
```

Note: If both of the attributes `triggerChecklist` and `triggerAVPs` are configured, then forwarding behavior happens ONLY when the incoming Accounting Request message matches both attributes.

All forwarding group will be checked to find the matched remote servers.

Note: The configuration takes effect after Radius AAA server restarts.



10 Configuring MSISDN Authorization for Home AAA Server

To configure MSISDN Authorization for Home AAA Server, perform the following procedures:

- Section 3.1 Configuring AAA Server Type on page 5
- Section 3.2 Configuring Home AAA Server Network Relationship on page 6
- Section 3.3 Configuring Radius Stack Interface on page 8
- Section 10.1 Configuring MSISDN AuthSelector on page 65
- The following topics are performed when CUDB is not used:
 - Section 10.2 Creating Users for MSISDN Authorization on page 66
 - Section 10.3 Configuring Policies for Authorization on page 67
 - Section 10.4 Configuring User Group for Authorization on page 67
 - Section 10.5 Linking User and Policy on page 68

10.1 Configuring MSISDN AuthSelector

MSISDN AuthSelector is not configured by default after IPWorks Radius AAA installation. The Radius AAA uses this selector function module to respond and handle the coming request which only contains Calling-Station-ID and without any other Trigger AVPs. For more information about Trigger AVPs, refer to the table in section Architecture of IPWorks Generic AAA Function Overview.

Following is an example to configure the MSISDN AuthSelector in the MO AuthSelectorMgr:

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksRad
(AuthSelectorMgr=1)> configure
(config-AuthSelectorMgr=1)>AuthSelector=msisdn-authorization
(config-AuthSelector=msisdn-authorization)>isNeedAuthorization=true
(config-AuthSelector=msisdn-authorization)>rank=50
(config-AuthSelector=msisdn-authorization)>triggerAVPs="Calling-Station-Id="
(config-AuthSelector=msisdn-authorization)>commit
(AuthSelector=msisdn-authorization)>
Confirm that the new selector is created successfully.
(AuthSelector=msisdn-authorization)>show -v
AuthSelector=msisdn-authorization
authSelectorId="msisdn-authorization"
```



```
isNeedAuthorization=true
rank=50
triggerAVPs
    "User-Name = *"
    "Calling-Station-Id=*"

```

Note: The configuration takes effect after Radius AAA server restarts.

10.2 Creating Users for MSISDN Authorization

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

The parameter username has the following meanings:

- The username.
- The access loop ID (Access Circuit ID) value for Fixed Access IPoE user.

Note: AAA is expected to receive the Access loop ID in AVP User-Name.

For more information, refer to the Section AAAUser in IPWorks AAA Parameter Description.

The ways to create the users are same for different user.

Create users according to the following examples through IPWorks CLI:

1. Execute the following command.

```
IPWorks> create AAAUser -set username=<username>;password=<password>
```

For example:

```
IPWorks> create AAAUser -set username=9876543210;password=123456789A123456789A12
```

```
1 object(s) created.
```

2. Confirm that the new users are created successfully.

```
IPWorks> list AAAUser
```

```
[AAAUser 9876543210]
Username: 9876543210
Password: *****
AuthMethod: NONE
IPAllocType: 0
IPAllocValue: 0
IPv6PrefixAllocType: 0
IPv6PrefixAllocValue: 0
```



Note: In MSISDN Authorization, the Username must be the same as Calling-Station-Id.

10.3 Configuring Policies for Authorization

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

1. Create one policy.

```
IPWorks> create AAPolicy AAPolicy0 -set checklist="Calling
-Station-Id=9876543210";replylist="Framed-Route=10.10.10.0,
SN-Rad-APN-Name=test1.com"
```

```
1 object(s) created.
```

2. Display the newly created policies.

```
IPWorks> list AAPolicy
[AAPolicy aaapolicy0]
Name: AAPolicy0
Checklist: Calling-Station-Id = 9876543210
Replylist: Framed-Route = 10.10.10.0, SN-Rad-APN-Name = test1.com
```

The checklist of AAPolicy0 is used to check the incoming packet AVPs, and to ensure that the value of attribute Calling-Station-Id is username. If the check result is True, the AVPs in Replylist will be appended to the response packet. Calling-Station-Id with the value in the request packet will be appended.

In MSISDN Authorization, the Checklist must be the same as Calling-Station-Id. The Replylist supports additional AVPs as SN-Rad-APN-Name, SN1-Rad-APN-Name, and HW-Requested-APN.

10.4 Configuring User Group for Authorization

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

1. Create a user group named MSISDN-Group.

```
IPWorks> create AAAUserGroup -set name=MSISDN-Group
```

```
1 object(s) created.
```

2. Modify the newly created user group to set policies AAPolicy0.

```
IPWorks> modify AAAUserGroup MSISDN-Group -set policy=AAAPoli
cy0
```



Working on 1 object(s).

3. Display the newly created user group.

```
IPWorks> list AAAUserGroup
[AAAUserGroup msisdn-group]
  Name: MSISDN-Group
  Policy: AAPolicy0
```

10.5 Linking User and Policy

If CUDB is configured, skip this procedure. For detailed information on CUDB configuration, refer to PG documents.

The AAA user 9876543210 is directly linked to the policies. The policies will be checked for the user in the alphabetical sequence in attribute policy. The first policy checked with TRUE is selected for authentication or authorization.

1. Modify the AAA user to set the policies.

```
IPWorks> modify AAAUser 9876543210 -set policy=AAPolicy0
Working on 1 object(s).
1 object(s) were updated.
```

2. Display the modified AAA user AAA-Test.

```
IPWorks> list AAAUser 9876543210
[AAAUser 9876543210]
  Username: 9876543210
  Password: *****
  AuthMethod: NONE
  Policy: AAPolicy0
  IPAllocType: 0
  IPAllocValue: 0
  IPv6PrefixAllocType: 0
  IPv6PrefixAllocValue: 0
```

3. The AAA user 9876543210 is indirectly linked to policy through the user group. The policies will be checked for the user in the alphabetical sequence in attribute policy of user group MSISDN-Group. The first policy checked with TRUE is selected for authentication or authorization.

```
IPWorks> modify AAAUser 9876543210 -set groupname=MSISDN-Group
Working on 1 object(s).
1 object(s) were updated.
```

4. Display the modified AAA user IMS-User.



```
IPWorks> list AAAUser 9876543210
[AAAUser 9876543210]
  Username: 9876543210
  Password: *****
  AuthMethod: NONE
  GroupName: MSISDN-Group
  Policy: AAPolicy0
  IPAllocType: 0
  IPAllocValue: 0
  IPv6PrefixAllocType: 0
  IPv6PrefixAllocValue: 0
```

If the user is directly linked to the policy, that is, attribute `policy` of the user is set, the policy of the user group will not impact AAA user policy selection strategy. When there are multiple directly linked policies for the user, the first policy whose checklist is evaluated as `TRUE` will be selected.

If the user is indirectly linked to the policy through the user group, the policy selection strategy is the same as that for the directly linked policy in one user group, that means the first policy whose checklist is evaluated as `TRUE` will be selected.

Note: It is strongly recommended to set policies mutually exclusive for one user or user group.





11 Radius AAA Operations

This section provides the procedure for common Radius AAA operations.

11.1 Restarting Radius AAA Server

IPWorks provides mechanisms for controlling the AAA server once it is installed and operating. You can use the `ipw-ctr` command to start or stop the server directly from the system where the server is in operation.

- Log on SC node (SC-1 or SC-2).

```
# ssh <Username>@<MIP_OAM_IP>
```

- Restart Radius Stack.

```
# ipw-ctr restart aaa_radius_stack <PL hostname>
```

- Restart Radius Backend.

```
# ipw-ctr restart aaa_radius_backend <PL hostname>
```

- Restart CSV Engine.

```
# ipw-ctr restart csvengine <SC hostname>
```

11.2 Viewing Server Logs

The AAA server allows the viewing of logs.

The AAA server log is configured by changing the attribute level of the MO `IPWorksLog`.

1. Log on to the ECLI.

```
# ssh <username>@<MIP_OAM_IP> -t -s cli
```

2. Enter the configuration mode.

```
>configure
```

3. Set the value of attribute level in the MO `IPWorksLog`. Choose the right server and process. The following is the example to change the AAA server log level on PL-3.

```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorks
AAARoot=1,IPWorksAAACCommonRoot=1,
AAAServer=PL-3,LogManagement=1,IPWorksLog=AAA_RADIUS_BACKEND
(config-IPWorksLog=AAA_RADIUS_BACKEND)>level=LOG_LEVEL_INFO
(config-IPWorksLog=AAA_RADIUS_BACKEND)>commit
```



```
>dn ManagedElement=<Node Name>,IpworksFunction=1,IPWorksAAARoot=1,IPWorksAAAC
AAAServer=PL-3,LogManagement=1,IPWorksLog=AAA_RADIUS_STACK
(config-IPWorksLog=AAA_RADIUS_STACK)>level=LOG_LEVEL_INFO
(config-IPWorksLog=AAA_RADIUS_STACK)>commit
```

4. View the logs in the related log file.

All logs are stored in the directory `/cluster/storage/no-backup/ipworks/logs`. The Radius AAA related log file is `/cluster/storage/no-backup/ipworks/logs/<hostname>/aaa_radius_<service>.log`

Where:

- The `<hostname>` represents the PL hostname that holds the AAA server.
- The `<process>` represents the process of Radius AAA server. Its value is backend or stack.

For example, you can view the AAA server logs on PL-3 by using the following command:

```
# tailf /cluster/storage/no-backup/ipworks/logs/PL-3/aaa_radiu
s_backend.log
```

```
# tailf /cluster/storage/no-backup/ipworks/logs/PL-3/aaa_rad
ius_stack.log
```



12 Appendix A: /etc/ipworks/aaa_radius/aaa_wifi_data.xml

12.1 Configuration Parameters

The following tables describes the attributes of each element:

Table 4 APN-Prefix

Element	Attribute	Type	Value Limitation	Unit
APN-Prefix ⁽¹⁾	S2A-Prefix	String	S2A-Prefix is prefix name aligned with HLR to identify APN retrieved from HLR is s2a, default value is s2a.	-
-	LBO-Prefix.	String	LBO-Prefix is prefix name aligned with HLR to identify APN retrieved from HLR is LBO, default value is lbo.	-
-	Prefered-Prefix	String	Either the value of S2A-Prefix or LBO-Prefix. Prefered-Prefix is set when APN from HLR has both S2a prefix and LBO prefix, default value is lbo.	-

(1) For using the APN mode, the configuration of APN-Prefix is basic and mandatory.

When the Trusted Wi-Fi Support feature is applied, S2a-Prefix and Prefered-Prefix must be configured.



Table 5 S2a-GTP-Tunnel-Parameter

Element	Child-Element	Attribute	Type	Value Limitation	Unit
S2a-GTP-Tunnel-Parameter ⁽¹⁾	Bearer-Qos	Pre-Emption-Vulnerability	Integer	Either 0 or 1	—
	—	Priority-Level	Integer	Between 1 and 15	—
	—	Pre-Emption-Capability	Integer	Either 0 or 1	—
	—	Qos-Class-Identifier	Integer	Between 0 and 255	—
	—	Maximum-Bit-Rate-For-Uplink	Integer	Between 0 and 10,000,000,000	kbps
	—	Maximum-Bit-Rate-For-Downlink	Integer	Between 0 and 10,000,000,000	kbps
	—	Guaranteed-Bit-Rate-For-Uplink	Integer	Between 0 and 10,000,000,000	kbps
	—	Guaranteed-Bit-Rate-For-Downlink	Integer	Between 0 and 10,000,000,000	kbps
	APN	Name	String	—	—
	—	APN-Restriction	Integer	Between 0 and 4	—
	—	APN-AMBR-Uplink	Integer	Between 0 and 4,294,967,295	kbps
	—	APN-AMBR-Downlink	Integer	Between 0 and 4,294,967,295	kbps
	—	PDN-GW-Address	String	The IP address list consists of at least two either IPv4 or IPv6 IP addresses. Both the Primary and Secondary PDN-GW Address select the IP from this IP address list according to a Round-Robin algorithm. ⁽²⁾⁽³⁾	—

(1) S2a-GTP-Tunnel-Parameter is used for the FeatureName-Trusted-WiFi-Support feature.

(2) Do not mix the IPv4 and IPv6 IP addresses in one IP address list. Separate the IP address by a comma.

(3) If PDN-GW-Address is set as "0.0.0.0,0.0.0.0", Wi-Fi GW gets PDN-GW addresses for the APN through DNS resolution.

Table 6 Enhanced-Wifi-Authorization-Parameter (1)

Element	Child-Element
Enhanced-Wifi-Authorization	S2A-APN-List
	APN-Blacklist



Table 7 Enhanced-Wifi-Authorization-Parameter (2)

Element	Child Element	Attribute	Type	Value Limitation
S2A-APN-List	S2A-APN	Name	String	<p>The value should be not more than 70 characters. For more information, refer to 3GPP TS 23.003.</p> <p>Name is configured as same as HLR saved apn-name.</p> <p>The value of APN Name in S2A-APN-List MUST match the APN-List.</p>
APN-Blacklist	Blacklist-APN	Name	String	<p>The value should be not more than 70 characters. For more information, refer to 3GPP TS 23.003.</p> <p>Name is configured as same as HLR saved apn-name.</p> <p>The value of Blacklist-APN in APN-Blacklist matches all the replies from the HLR, the APNs will be barred.</p>

12.2 Example for Wi-Fi AAA

The following example shows the content of the file /etc/ipworks/aaa_radius/aaa_wifi_data.xml.



```

<?xml version="1.0" encoding="UTF-8"?>
<!--Make sure the value you configured is valid, or AAA Plugin EAP-AkaSim cannot start-->
<AAA_WiFi_Data>

  <APN-Prefix
    S2A-Prefix="s2a"
    LBO-Prefix="lbo"
    Prefered-Prefix="lbo"
  >
</APN-Prefix>

  <S2a-GTP-Tunnel-Parameter>
    <!--Pre-Emption-Vulnerability should be an integer value between 0 and 1.-->
    <!--Priority-Level should be an integer value between 1 and 15.-->
    <!--Pre-Emption-Capability should be an integer value between 0 and 1.-->
    <!--Qos-Class-Identifier should be an integer value between 0 and 255.-->
    <!--Maximum-Bit-Rate-For-Uplink should be an integer value between 0 and 1000000000 in kbps.-->
    <!--Maximum-Bit-Rate-For-Downlink should be an integer value between 0 and 1000000000 in kbps.-->
    <!--Guaranteed-Bit-Rate-For-Uplink should be an integer value between 0 and 1000000000 in kbps.-->
    <!--Guaranteed-Bit-Rate-For-Downlink should be an integer value between 0 and 1000000000 in kbps.-->
    <Bearer-Qos
      Pre-Emption-Vulnerability="1"
      Priority-Level="2"
      Pre-Emption-Capability="1"
      Qos-Class-Identifier="1"
      Maximum-Bit-Rate-For-Uplink="1000"
      Maximum-Bit-Rate-For-Downlink="2000"
      Guaranteed-Bit-Rate-For-Uplink="500"
      Guaranteed-Bit-Rate-For-Downlink="1000"
    >
    </Bearer-Qos>

    <!--APN-Restriction should be an integer value between 0 and 4.-->
    <!--APN-AMBR-Uplink should be an integer value between 0 and 4294967295 in kbps.-->
    <!--APN-AMBR-Downlink should be an integer value between 0 and 4294967295 in kbps.-->
    <!--PDN-GW-Address should be an ip list (containing at least two IPs) in which the ip address is seperated by
    <APN Name="cmnet.mnc000.mcc460.gprs"

      APN-Restriction="1"
      APN-AMBR-Uplink="1024"
      APN-AMBR-Downlink="10240"
      PDN-GW-Address="10.170.15.100,10.170.15.101,10.170.15.102"

    >
    </APN>
  </S2a-GTP-Tunnel-Parameter>

  <Enhanced-Wifi-Authorization>
    <!--S2A-APN-List saves the APN for s2a subscriber.-->
    <S2A-APN-List>
      <!--Name is configured as same as HLR saved apn-name.-->
      <!--The APN Name must be configured in the APN-List section above too.-->
      <S2A-APN Name="aaa.apn.com.cn"></S2A-APN>
    </S2A-APN-List>
    <!--APN-Blacklist saves the blacklist APNs which will be rejected.-->
    <APN-Blacklist>
      <!--Name is configured as same as HLR saved apn-name.-->
      <Blacklist-APN Name="coop.tim.it"></Blacklist-APN>
    </APN-Blacklist>

  </Enhanced-Wifi-Authorization>
</AAA_WiFi_Data>

```




Reference List

Ericsson Documents

- [1] Trademark Information
- [2] Typographic Conventions
- [3] Glossary of Terms and Acronyms
- [4] License Management
- [5] IPWorks Configuration Management
- [6] IPWorks AAA Parameter Description
- [7] Managed Object Model (MOM)
- [8] Command Line Interface User Guide for IPWorks SS
- [9] Ericsson Command-Line Interface User Guide
- [10] Configure SS7 for AAA
- [11] IPWorks Wi-Fi AAA Function Overview, 60/155 17-AVA 901 16 Uen
- [12] IPWorks Generic AAA Function Overview, 59/155 17-AVA 901 16 Uen