

Check Subscriber Access

• Checking Access, on page 1

Checking Access

When you are confident that the installation and configuration tasks are complete and processing properly, try running a small amount of test traffic, following it through the system. Here are three ways to ascertain correct process of access from a subscriber perspective.

Testing Subscriber Access with 00.testAccessRequest.sh

00.testAccessRequest.sh is a test script used to test subscriber access to the ISG and CPS system.

You can find **00.testAccessRequest.sh** in /opt/broadhop/installer/isg/troubleshooting directory on the CPS server.

To configure the subscriber used, edit the /opt/broadhop/installer/isg/troubleshooting/config.ini file.

Step 1 In the config.ini file, change the User-Name and Password fields.

Note You may need to change some of the other parameters in order to match your configuration. The other main attributes to change will be the NAS-IP-Address and Framed-IP-Address.

Step 2 Run the script from a command line. No arguments are necessary:

00.test Access Request.sh

Upon success, this output is displayed as follows:

Figure 1: 00.testAccessRequest.sh Output

Proot@lab:/opt/broadhop/installer/isg/troubleshooting	
[root@lab troubleshooting]# 00.testAccessRequest.sh	-
"NAS-IP-Address=192.168.181.29, Framed-IP-Address=70.1.2.43, User-Name=banner, User	
-Password=banner,NAS-Identifier=QA-Test,Calling-Station-Id=0200.0000.0001,Acct-S	
ession-Id=00000626" radclient 192.168.181.112:1812 auth aaacisco -r 1 -x	
Sending Access-Request of id 54 to 192.168.181.112 port 1812	
NAS-IP-Address = 192.168.181.29	
Framed-IP-Address = 70.1.2.43	
User-Name = "banner"	
User-Password = "banner"	
NAS-Identifier = "QA-Test"	
Calling-Station-Id = "0200.0000.0001"	
Acct-Session-Id = "00000626"	
rad_recv: Access-Accept packet from host 192.168.181.112 port 1812, id=54, lengt	
h=76	=
Cisco-Account-Info = "A2M-UP-DOWN"	
Cisco-AVPair = "accounting-list=QNS_ACCT_LIST"	-

Testing Subscriber Access with soapUI

This procedure tests end subscriber access to your system.

Step 1 Download soapUI from here: http://www.soapui.org/

You only need the freeware version (not the soapUI Pro).

- Step 2 Launch soapUI.
- Step 3 Right click on Projects and select New soapUIProject from the drop-down list.
- **Step 4** Name your project and enter into **Initial WSDL/WADL** the appropriate WSDL URL (you may have to replace the IP in display with your own IP) and select **OK**.

Figure 2: New soapUIProject

S New soapUI Project				
New soapUI Project Creates a new soapUI Project in this workspace				
Proj	iect Name:	Demo		
Initi	al WSDL/WADL:	http://199.166.181.112:8080/ua/wsdl/UnifiedApi.wsdl	Browse	
Crea	ate Requests:	Create sample requests for all operations?		
Crea	ate TestSuite:	Creates a TestSuite for the imported WSDL or WADL		
Crea	ate MockService:	Creates a Web Service Simulation of the imported WSDL		
Add	REST Service:	Opens dialog to create REST Service		
Rela	ative Paths:	$\hfill\square$ Stores all file paths in project relatively to project file (requires	save)	
Create Web TestCase: Creates a TestCase with a Web Recording session for functional web testing				
]		OK Cancel	

Step 5 In the tree, select **Demo** > **SubscriberBinding** > **GetSubscriber** > **Request 1** as shown in the following figure:

Figure 3: Request 1 Node

🖶 📵 Demo	
🖮 🎩 SubscriberBinding	
🕀 🤣 AddCredential	
AddCredentials	
🕀 👶 AuthenticateSubscriber	
🕀 👶 ChangeBalanceSubscriberId	
ChangeCredentialPassword	
ChangeCredentialUsername	
🕀 👶 ChangeRecurringRefreshDay	
🕀 🤣 CreateBalance	
CreateSubscriber	
CreateSubscribers	
🕀 🤣 CreateVoucher	
CreateVouchers	
🕀 🤣 Credit	
🕀 🤣 Debit	
🕀 🤣 DeleteBalance	
🕀 🤣 DeleteCredential	
DeleteCredentials	
🕀 🔅 DeleteQuota	
🕀 👶 DeleteSubscriber	
🕀 🤣 DeleteVoucher	
🖃 🤣 GetSubscriber	
Request 1	
🕀 🔁 KeepAlive	744
	10

Step 6 Select edit current.. to edit the end point. Enter the appropriate IP.

Figure 4: Request 1 XML File



Step 7 In the XML file:

- Replace ? in <typ:networkId>?</typ:networkId> with the appropriate credential or network ID.
- Replace ? in <typ:returnSessions>?</typ:returnSessions> with True.
- Replace ? in <typ:returnBalance>?</typ:returnBalance> with True.

Figure 5: Request 1 XML File



 Step 8
 Click the green arrow (underneath Request 1).

 Figure 6: Request 1 XML File



Step 9 Check the resulting XML output. Pay special attention to the relevant subscriber information.

Figure 7: XML Output

se:Envelo	pe xmlns:se="http://schemas.xmlsoap.org/soap/envelope/">
<se:bod< td=""><td>y></td></se:bod<>	y>
<get:< td=""><td>SubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types"></td></get:<>	SubscriberResponse xmlns="http://broadhop.com/unifiedapi/soap/types">
<	errorCode>0
<	errorMessage>Request completed successfully
<	subscriber>
	<id>4fb54d03e4b01e8478d309c2</id>
	<name></name>
	<fullname>Bruce Banner</fullname>
	<credential></credential>
	<networkid>banner</networkid>
	<pre><password>banner</password></pre>
	<credential></credential>
	<pre><networkid>0200.0000.0001</networkid></pre>
	<pre><expirationdate>2012-05-17T13:17:07.020-06:00</expirationdate></pre>
	<service></service>
	<code>SERVICE_A</code>
	<pre><enabled>true</enabled></pre>
	<session></session>
	<sessionkey></sessionkey>
	<code>UserIdKey</code>
	<primary>false</primary>
	<keyfield></keyfield>
	<code>userId</code>
	<value>banner</value>
	<sessionobject></sessionobject>
	<entry></entry>
	<string>tags</string>
8	t>