



## **Avaya Solution & Interoperability Test Lab**

---

# **Application Notes for Netas Avaya Utility Server (NAUS) with Avaya 9600 Series IP Deskphones - Issue 1.0**

### **Abstract**

These Application Notes describe the steps required to integrate Netas Avaya Utility Server (NAUS) with Avaya 9600 Series IP Deskphones. NAUS is a Java EE web application used to build applications using the Push Interface API. NAUS is capable of sending text messages to the top line and display sections of Avaya 9600 Series IP Deskphones, providing a LDAP database with people information that may be used as a directory by the phones, and provide information from the internet, such as weather and exchanges rates, using the deskphone browser. During the compliance test, a sample application was used to verify interoperability between NAUS and Avaya 9600 IP Deskphones using the Push Interface API.

Readers should pay attention to Section 2, in particular the scope of testing as outlined in Section 2.1 as well as the observations noted in Section 2.2, to ensure that their own use cases are adequately covered by this scope and results.

Information in these Application Notes has been obtained through DevConnect compliance testing and additional technical discussions. Testing was conducted via the DevConnect Program at the Avaya Solution and Interoperability Test Lab.

# 1. Introduction

These Application Notes describe the steps required to integrate Netas Avaya Utility Server (NAUS) with Avaya 9600 Series IP Deskphones. NAUS is a Java EE web application used to build applications using the Push Interface API. NAUS is capable of sending text messages to the top line and display sections of Avaya 9600 Series IP Deskphones, providing a LDAP database with people information that may be used as a directory by the phones, and provide information from the internet, such as weather and exchanges rates, using the deskphone browser. During the compliance test, a sample application was used to verify interoperability between NAUS and Avaya 9600 IP Deskphones using the Push Interface API.

The sample NAUS application provided the following functionality:

1. From the NAUS web console, send text messages to the top line and display sections of Avaya 9600 IP Deskphones. All messages are stored in the database with send time and send information.
2. Avaya 9600 IP Deskphones can use its browser to search a directory for a person in a department, maintained in a LDAP database, and then call that person.
3. Avaya 9600 IP Deskphones can use its browser to weather and exchange rate information from the internet. Weather and exchange rate data are retrieved from 3<sup>rd</sup> party websites.

NAUS would automatically send text messages to all deskphones that subscribe to it. Avaya deskphones register with NAUS during startup via subscription requests and NAUS sends re-subscribe requests to deskphones periodically. Deskphones that do not respond to the subscribe request are removed from the system. The directory in the LDAP database is created manually.

## 2. General Test Approach and Test Results

This section details the general approach to the testing, what was covered, and results of the testing. If the testing was successfully concluded but it was necessary to implement workarounds or certain non-critical features did not work, it should be noted in **Section 2.2**.

DevConnect Compliance Testing is conducted jointly by Avaya and DevConnect members. The jointly-defined test plan focuses on exercising APIs and/or standards-based interfaces pertinent to the interoperability of the tested products and their functionalities. DevConnect Compliance Testing is not intended to substitute full product performance or feature testing performed by DevConnect members, nor is it to be construed as an endorsement by Avaya of the suitability or completeness of a DevConnect member's solution.

The interoperability compliance test included feature and serviceability testing. Feature testing focused on sending text messages from the NAUS web console to Avaya 9600 IP Deskphones, deskphones searching a directory and retrieving internet data from its browser using the Push API.

The serviceability testing focused on verifying that NAUS comes back into service after re-connecting the Ethernet cable or rebooting the server.

## 2.1. Interoperability Compliance Testing

Interoperability compliance testing covered the following features and functionality:

- Subscribed Avaya 9600 IP Deskphones to NAUS using the Push Interface API.
- Verified Avaya 9600 IP Deskphones running H.323 and SIP software, including the following phone types: 9608, 9611, 9620, 9621, 9641, and 9650.
- From the NAUS web console, sent text messages to the top line and display sections of Avaya 9600 IP Deskphones.
- Verified that a maximum of 40 characters can be included for top line messages and a maximum of 250 characters can be included for display messages.
- Verified that all messages sent were stored in the NAUS database.
- Verified that deskphones that do not respond to subscribe requests are removed from the system after a configured time interval.
- Avaya 9600 IP Deskphones can use its browser to perform the following functions:
  - Search a directory for a person and then call that person. The directory was maintained in a LDAP database that was created manually.
  - Retrieve weather and exchange rate data from 3<sup>rd</sup> party websites.
- Proper system recovery after a restart of the NAUS server and loss of IP connectivity.

## 2.2. Test Results

All test cases passed with the following observation:

- Avaya SIP deskphones do not scroll through a long top line message. At the end of the top line message, it displays the ellipsis (...) indicating that the message is longer than what is displayed.
- Avaya SIP deskphones display an extra '\$' when that symbol is included in a top line message. For example, '\$500' will be displayed as '\$\$500'.
- Avaya SIP deskphones do not display the dollar sign (\$) for display messages.
- Avaya 96x1 SIP deskphones do not display images on its display (e.g., an image of a cloud as it relates to the weather would not be displayed).

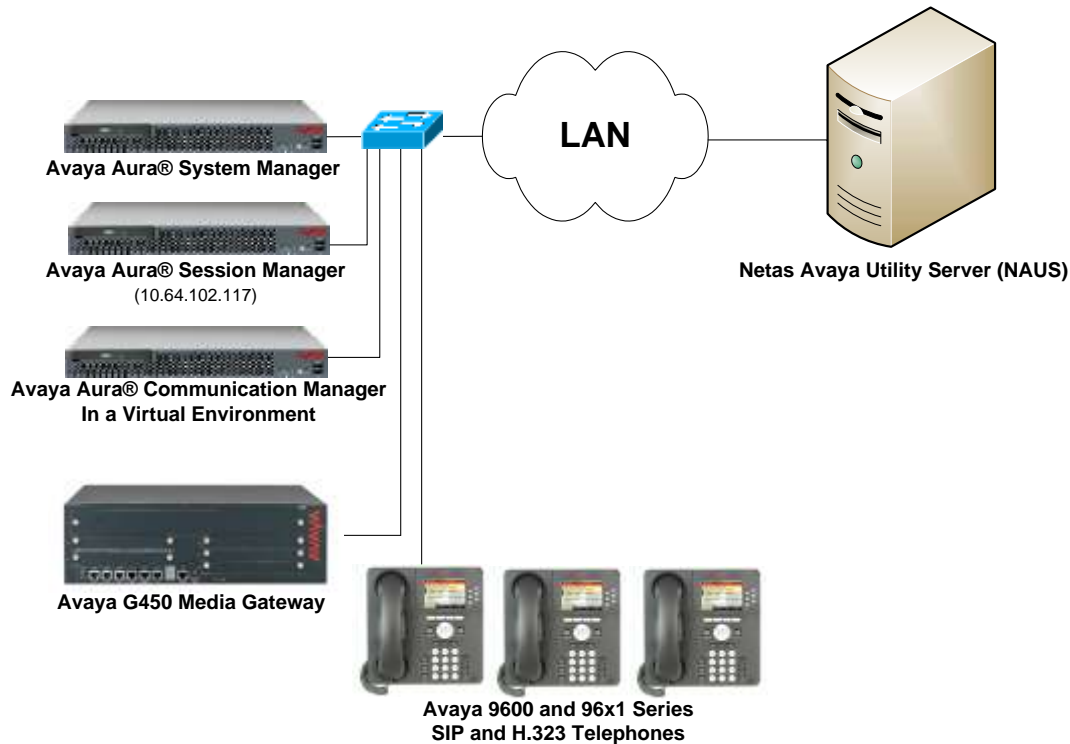
## 2.3. Support

For technical support on Netas Avaya Utility Server (NAUS), contact Netas support via phone, email, or website.

- **Phone:** +90 (216) 522 20 00
- **Email:** [naus@netas.com.tr](mailto:naus@netas.com.tr)
- **Web:** <http://www.netas.com.tr/en/services/maintenance-and-support-services/913>

### 3. Reference Configuration

**Figure 1** illustrates a sample configuration consisting of NAUS with Avaya Aura® Communication Manager, Avaya Aura® Session Manager, and 9600/96x1 IP Series IP Telephones. NAUS was the trusted push server to which the Avaya IP deskphones subscribed. Avaya Aura® System Manager was used to configure Session Manager. The test configuration was also connected to the internet.



**Figure 1: NAUS with Avaya 9600/96x1 Series IP Deskphones**

## 4. Equipment and Software Validated

The following equipment and software were used for the sample configuration provided:

Equipment/Software	Release/Version
Avaya Aura® Communication Manager in a Virtual Environment with an Avaya G450 Media Gateway	7.0.0.2 SP 2 (R017x.00.0.441.0 with Patch 22684)
Avaya Aura® System Manager	7.0 (Build No. 7.0.0.016266-7.0.9.912 Software Update Revision No: 7.0.0.0.3929)
Avaya Aura® Session Manager	7.0.0.0.700007
Avaya Aura® Messaging	6.3.2 SP 2 Patch 3
Avaya 9600 Series IP Deskphones <ul style="list-style-type: none"><li>▪ 9620C</li><li>▪ 9650</li></ul>	S3.260A (H.323) 2.6.15.0 (SIP)
Avaya 96x1 Series IP Deskphones <ul style="list-style-type: none"><li>▪ 9608</li><li>▪ 9611G</li><li>▪ 9621G</li><li>▪ 9641G</li></ul>	6.6115 (H.323) 7.0.0.39 (SIP)
Netas Avaya Utility Server (NAUS)	V1.1

## 5. Configure Avaya Aura® Communication Manager

This section covers the station configuration for the Avaya 9600 IP Deskphones that will subscribe to NAUS. The configuration is performed via the System Access Terminal (SAT) on Communication Manager or via Avaya Aura® System Manager for SIP stations.

### 5.1. Configure a Station for Avaya 9600 Series H.323 Deskphone

Use the **add station** command to create a station for the 9608 IP deskphone (other phone types are supported). Set the **Type** field to the station type to be emulated. In this example, *9608* was used. Set the **Port** field to *IP* and configure a **Security Code** as that password to be used by the Avaya deskphone to log in. Repeat this procedure for each H.323 station that will subscribe to NAUS.

```
add station 77302                                     Page 1 of 5
                                                    STATION
Extension: 77302                                     Lock Messages? n          BCC: 0
  Type: 9608                                       Security Code: 1234      TN: 1
  Port: IP                                       Coverage Path 1:          COR: 1
  Name: IP 77302                                   Coverage Path 2:          COS: 1
                                                    Hunt-to Station:          Tests? y
STATION OPTIONS
                                                    Time of Day Lock Table:
  Loss Group: 19                                   Personalized Ringing Pattern: 1
                                                    Message Lamp Ext: 77302
  Speakerphone: 2-way                             Mute Button Enabled? y
  Display Language: english                       Button Modules: 0
Survivable GK Node Name:
  Survivable COR: internal                         Media Complex Ext:
  Survivable Trunk Dest? y                        IP SoftPhone? n
                                                    IP Video? n
Short/Prefixed Registration Allowed: default
                                                    Customizable Labels? y
```

## 5.2. Configure a Station for Avaya 9600 Series SIP Deskphone

The SIP station was configured automatically through Avaya Aura® System Manager. Use the **display station** command to view the station for the 9608 IP deskphone (other phone types are supported). The **Station Type** was set to *9608SIP* and a descriptive **Name** was also provided. Use the default values for the other fields on **Page 1**. Repeat this procedure for each SIP station that will subscribe to NAUS.

display station 78002		Page	1 of	6
		STATION		
<b>Extension:</b>	78002	Lock Messages?	n	BCC: 0
<b>Type:</b>	9608SIP	Security Code:		TN: 1
Port:	S00005	Coverage Path 1:		COR: 1
Name:	78002, SIP	Coverage Path 2:		COS: 1
		Hunt-to Station:		
STATION OPTIONS		Time of Day Lock Table:		
	Loss Group: 19	Message Lamp Ext: 78002		
	Display Language: english	Button Modules: 0		
	Survivable COR: internal	IP SoftPhone? n		
	Survivable Trunk Dest? y	IP Video? n		

## 6. Configure Avaya 9600 Series IP Deskphones

NAUS uses the Avaya Push Interface API to send text messages and provide browser functionality to Avaya IP deskphones. To authorize NAUS as a Trusted Push Server, the Push Interface settings in the **46xxsettings.txt** file must be configured properly. The Avaya IP deskphones must retrieve those settings during start-up.

Open the 46xxsettings.txt file and locate the PUSH INTERFACE SETTINGS. Edit the following parameters as follows:

- SET TPSLIST 192.168.100.205
- SET SUBSCRIBELIST <http://192.168.100.205/naus/subscribePhone.xml>

**Note:** Use the appropriate NAUS IP address and URL that corresponds to the customer's network.

Below are the Push Interface settings in the **46xxsettings.txt** file.

```
##### PUSH INTERFACE SETTINGS #####
##
## These settings are used to administer the Push interface.
## These parameters are not supported on 16cc SIP phones.
##
## The list of all the Trusted Push Servers.
## If set to "/", all servers are allowed.
## If set to null or blank, Push is disabled.
SET TPSLIST 192.168.100.205
##
## The list of all the Subscription Servers.
SET SUBSCRIBELIST http://192.168.100.205/naus/subscribePhone.xml
##
##
```



Under the 46xx PUSH INTERFACE SETTINGS section, configure the following additional parameters:

- SET PUSHPORT 80
- SET PUSHCAP 222
- SET WMLHOME <http://192.168.100.205/naus/index-wml.xml>

**Note:** The WMLHOME parameter may be configured on a per phone type basis in the 46xxsettings.txt file. Make sure that the WMLHOME parameter is set under each phone type in the 46xxsettings.txt file or set the parameter in a section of the file where it would apply to all phone types as shown below. In other words, if the WMLHOME parameter is only set for 9650 phone type, the parameter would not be set for all other phone types.

```
##### 46xx PUSH INTERFACE SETTINGS #####
##
## These settings are used to administer the Push interface
##
## The TCP port number for the telephone's HTTP server.
## (80-65535). The default is 80.
SET PUSHPORT 80
##
## Push capabilities settings. PUSHCAP consists of 3 bits
## (each 0, 1, or 2). The rightmost bit controls the Top
## Line push mode. The middle bit controls the Web
## Browser push mode. The leftmost bit controls the Audio
## push mode.
##
## When PUSHCAP is set to 000, all push modes are disabled
##
## When PUSHCAP is set to 111, barge in only is allowed in
## all push modes.
##
## When PUSHCAP is set to 222, both barge in and normal
## pushes are allowed in all push modes.
SET PUSHCAP 222
SET WMLHOME http://192.168.100.205/naus/index-wml.xml
##
```

After all the updates are made, save the **46xxsettings.txt** file.

## 7. Configure NAUS

The executable file of NAUS is a .war file that is deployed in an application server. The name of the file is **naus.war**. For the compliance test, Glassfish, an open-source application server was used. The application uses a MySQL database to maintain the directory. The installation and configuration is performed by Netas. End users may use the NAUS web console to send top line and display messages to the deskphones.

### 7.1. parameters.properties Configuration File

NAUS configuration is also contained in the **parameters.properties** file located in the /glassfish-4.0/glassfish4/glassfish/domains/domain1/applications/naus/WEB-INF/classes/configs directory. Below is the content of the **parameters.properties** file used for the compliance test.

```
#Application uses the IP address when it sends push messages.
#It is the IP in the 46xxsettings.txt (SET TPSLIST http://10.254.63.12/)
SERVER_DOMAIN=192.168.100.205

#The port of application in server
SERVER_PORT=80

#The port that is listened by phones for push messages
#SET PUSHPORT 80 in 46xxsettings.txt
PUSH_PORT=80

#The prefix of telephone numbers
#In LDAP,telephone numbers are 4 digit. The phone calls the number 62xxxx if it is 62.
PHONE_PREFIX=

#Number of thread used when sending push messages
NUMBER_OF_PUSH_THREAD =5

#A phone is removed from the system when it doesn't answer subscription request during
this period.
EXPIRE_TIME_IN_MINUTES=60

#The duration between 2 subscription request from server to phones
CLIENT_UPDATE_RATE_IN_MINUTES=30

#The duration between updating the weather condition information in the server
WEATHER_UPDATE_RATE_IN_MINUTES=60

#The duration between updating the exchange rate information in the server
EXCHANGE_UPDATE_RATE_IN_MINUTES=60

#The name of city for users
WEATHER_CITY=Istanbul

# the latitude value used in request to http://forecast.io/
WEATHER_API_LATITUDE=41.0186

# the longitude value used in request to http://forecast.io/
WEATHER_API_LONGITUDE=28.9647

#The api key used in http://forecast.io/ request
WEATHER_API_KEY=1d6eb9f46371797322dda4aac2930778
```

```
#LDAP is used for authentication and people search in this application
```

```
#LDAP URL
```

```
LDAP_PROVIDER_URL=LDAP://192.168.100.205:389/
```

```
#LDAP Name
```

```
LDAP_SECURITY_BASE=DC=netas,DC=lab,DC=avaya,DC=com
```

```
#LDAP authentication type
```

```
LDAP_SECURITY_AUTHENTICATION=simple
```

```
#The DN of account used by application
```

```
LDAP_SECURITY_PRINCIPAL=CN=AvayaClient,DC=netas,DC=lab,DC=avaya,DC=com
```

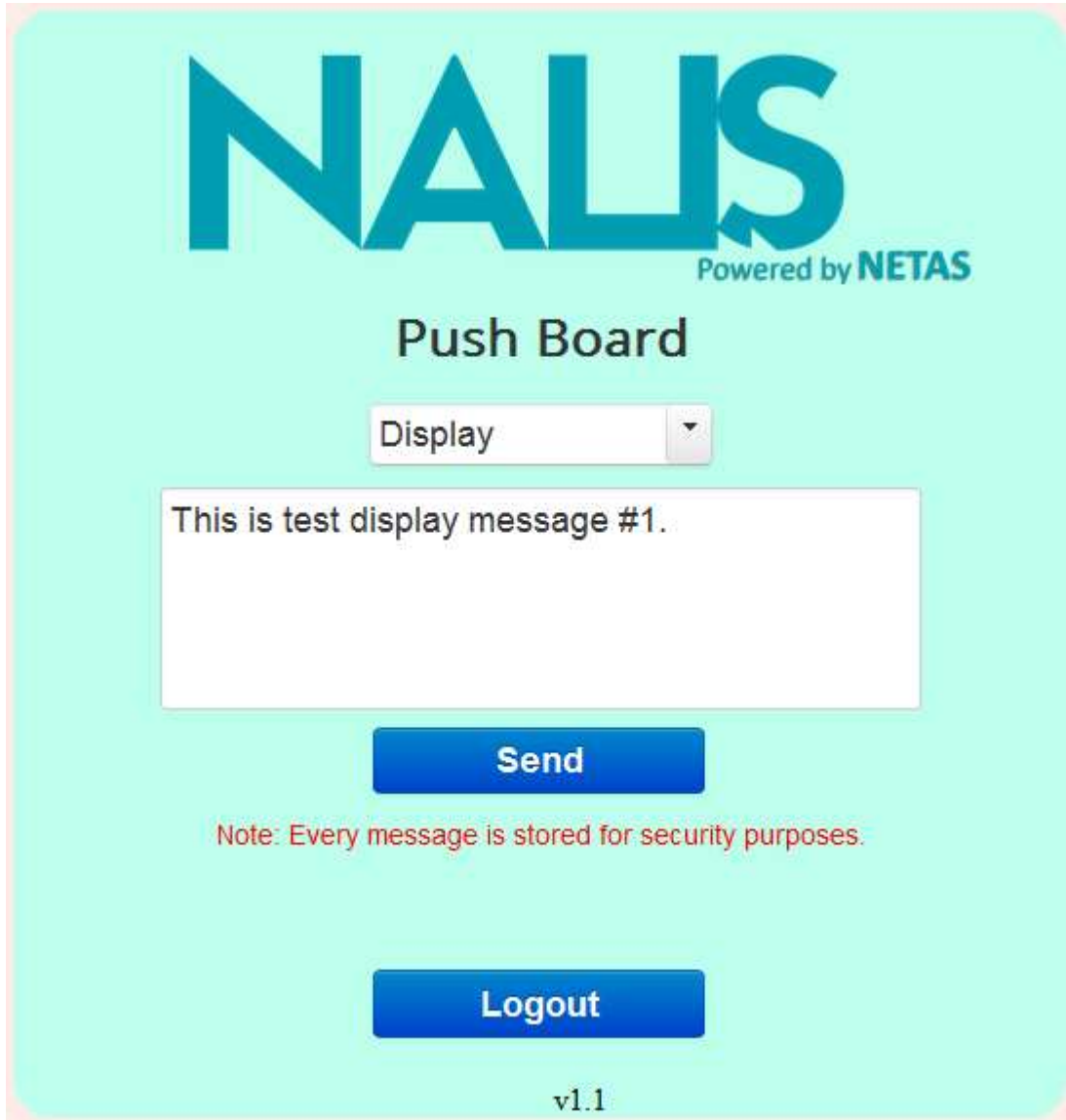
```
#The password of the account set in LDAP_SECURITY_PRINCIPAL
```

```
LDAP_SECURITY_CREDENTIALS=avaya23*A
```

## 8. Verification Steps

This section provides the tests that can be performed to verify proper configuration of Avaya 9600 IP Deskphones and NAUS.

1. In an internet browser, enter *localhost:8080* and log into the NAUS web console. Select *Topline* or *Display* in the **Push Type** drop-down field and enter a message as shown below. Press **Send**.



The screenshot shows the NAUS Push Board web interface. At the top is the NAUS logo with the text "Powered by NETAS" to its right. Below the logo is the title "Push Board". Underneath the title is a drop-down menu currently set to "Display". Below the menu is a text input field containing the message "This is test display message #1.". Below the input field is a blue "Send" button. Below the button is a red note that reads "Note: Every message is stored for security purposes.". Below the note is a blue "Logout" button. At the bottom center of the interface is the version number "v1.1".

2. Verify that all deskphones subscribed to NAUS received the text message as shown below.



3. From the deskphones browser, choose an option to search the people directory and then place a call to a person in the directory or retrieve data from the internet (e.g., weather or exchange rates). Verify that the person is found in the directory and that the call is established successfully or that internet data is displayed on the deskphone.

## 9. Conclusion

These Application Notes describe the steps required to integrate Netas Avaya Utility Server (NAUS) with Avaya 9600 Series IP Deskphones. NAUS delivered text messages to Avaya 9600 and 96x1 IP Deskphones using the Push Interface API. In addition, Avaya 9600 IP Deskphones were able to search a people directory and retrieve data from 3<sup>rd</sup> party websites using its browser. All test cases passed with observations noted in **Section 2.2**.

## 10. References

This section references the Avaya documentation relevant to these Application Notes. The Avaya product documentation is available at <http://support.avaya.com>.

- [1] *Avaya one-X® Deskphone Edition for 9600 Series IP Telephones Application Programmer Interface (API) Guide*, Issue 6, August 2012, Document Number 16-600888.
- [2] *NAUS Application*, provided by Netas.

---

**©2016 Avaya Inc. All Rights Reserved.**

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at [devconnect@avaya.com](mailto:devconnect@avaya.com).