Avaya Solution & Interoperability Test Lab

Configuring the Service Components Conferencing Sample Application with Ubiquity Developer Edition, Avaya Communication Manager, and Avaya SIP Enablement Services – Issue 1.0

Abstract

These Application Notes describe the steps to configure the Service Components Conferencing sample application to work with an environment comprised of Ubiquity Developer Edition (UDE), Avaya Communication Manager, and Avaya SIP Enablement Services.

The Conferencing application is one of several sample applications available with UDE. This application sets up and manages conferences including adding and removing Avaya endpoints as participants. The Conferencing application also supports playing audio messages, using the Cantata SnowShore IP Media Server, to individual endpoints or to multiple endpoints that are conferenced together. The Conferencing application is installed with the Service Components Start-Up Pack.
1. Introduction

These Application Notes describe the steps to configure the Service Components Conferencing sample application to work with an environment comprised of Ubiquity Developer Edition (UDE), Avaya Communication Manager, and Avaya SIP Enablement Services (SES).

1.1. Background

In the sample configuration, the Avaya SIP Application Server (A/S) environment consists of UDE as the base platform (UDE is the developer version of the Avaya SIP A/S). The Service Oriented Object Framework (SOOF), also known as SIP Service Oriented Architecture (SOA), is a SOA-based application development and deployment framework that is deployed on top of UDE. SOOF includes a suite of re-usable SIP Service Components.

Several sample applications are available to demonstrate the types of applications and services that can be created using the services provided with the Service Components. The Conferencing application is one of these sample applications. The Conferencing sample application described in these Application Notes is installed with the Service Components Start-Up Pack [6]. This application uses several services supported by the Service Components:

- Call service: make and receive calls
- Conferencing service: set up and manage conferences
- Interactive Voice Response (IVR) service: play audio messages

UDE uses the Cantata SnowShore IP Media Server as a conference bridge and to play audio messages. NOTE: The conferencing and IVR services of Avaya Communication Manager are not tested as part of this sample configuration.

In the sample configuration, UDE communicates with Avaya Communication Manager via a SIP trunk to Avaya SES. UDE is configured as an adjunct to Avaya SES. The Conferencing application was used to set up and manage conferences. Endpoints were added to conferences either by being dialed by the application or the endpoints dialed into the conference directly. To allow endpoints to dial into the Conferencing service, a hunt group was configured in Avaya Communication Manager to associate a “dial-in number” with the Conferencing service on UDE. Using the Cantata SnowShore IP Media Server, the Conferencing application also was used to play audio messages to individual endpoints and to multiple endpoints that were conferenced together.

These Application Notes focus on the following:

- The configuration of the SIP trunks between Avaya Communication Manager and Avaya SES for routing calls to and from UDE
- The configuration of the hunt group to allow endpoints to dial into the Conferencing service
- The configuration of Avaya SES to support UDE as an adjunct system
- The configuration of UDE to support the Conferencing service
- The configuration of the Conferencing sample application
For administration of endpoint telephones and general administration of Avaya Communication Manager and Avaya SES, refer to the appropriate documentation listed in Section 9.

1.2. Network Configuration

These Application Notes present a sample configuration for the network shown in Figure 1. For the sample configuration, Avaya Communication Manager is running on the Avaya S8300 Server with an Avaya G700 Media Gateway. The sample configuration described in these Application Notes is applicable to other Avaya servers and gateways.

SIP trunks are used to connect Avaya Communication Manager and UDE via Avaya SES. All calls established by or with the Conferencing service are carried over these SIP trunks. A connection to the public switched telephone network (PSTN) is established using a DS1 trunk. As this is not the focus of these Application Notes, the configuration of the PSTN trunk is not covered.

As shown in Figure 1 and listed in Table 1, there are several Avaya Digital, H.323, and SIP telephones that are connected to or registered with Avaya Communication Manager or Avaya SES. Additionally, there are PSTN telephones connected to the configuration via the PSTN trunk. With the exception of the Digital and the PSTN telephones, all components are connected to a single Avaya C364T-PWR Converged Stackable Switch.
Figure 1: Network Configuration Diagram
## 2. Equipment and Software Validated

The following components were used for the sample configuration provided:

<table>
<thead>
<tr>
<th>Component</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avaya S8300 Server with Avaya G700 Media Gateway</td>
<td>Avaya Communication Manager 4.0.1 Load 731.2 (R014x.00.1.731.2)</td>
</tr>
<tr>
<td>Avaya SIP Enablement Services</td>
<td>4.0, Load 33.6 (SES-4.0.0.0-033.6)</td>
</tr>
<tr>
<td>Avaya SIP Application Server environment Ubiquity Developer Edition</td>
<td>1.3.8</td>
</tr>
<tr>
<td>Ubiquity Development Studio</td>
<td>1.3.6</td>
</tr>
<tr>
<td>Service Oriented Object Framework</td>
<td>1.1.5</td>
</tr>
<tr>
<td>Service Components</td>
<td>1.5.2</td>
</tr>
<tr>
<td>Avaya 4610SW IP Telephone</td>
<td>2.8 (H.323)</td>
</tr>
<tr>
<td></td>
<td>3.1.0 (Avaya Quick Edition)</td>
</tr>
<tr>
<td>Avaya 4621SW IP Telephone</td>
<td>2.8 (H.323)</td>
</tr>
<tr>
<td></td>
<td>2.2.2 (SIP)</td>
</tr>
<tr>
<td></td>
<td>3.1.0 (Avaya Quick Edition)</td>
</tr>
<tr>
<td>Avaya 9620 IP Telephone</td>
<td>1.5 (H.323) (Avaya one-X™ Deskphone Edition)</td>
</tr>
<tr>
<td></td>
<td>1.0.13.1 (5) (Avaya one-X Deskphone SIP)</td>
</tr>
<tr>
<td>Avaya 9630 IP Telephone</td>
<td>1.5 (H.323) (Avaya one-X Deskphone Edition)</td>
</tr>
<tr>
<td></td>
<td>1.0.13.1 (5) (Avaya one-X Deskphone SIP)</td>
</tr>
<tr>
<td>Avaya 6408D+ Digital Telephone</td>
<td>N/A</td>
</tr>
<tr>
<td>Avaya C364T-PWR Converged Stackable Switch</td>
<td>4.5.14</td>
</tr>
<tr>
<td>Eclipse SDK</td>
<td>3.2.1</td>
</tr>
<tr>
<td>Sun Microsystems Java</td>
<td>1.5.0</td>
</tr>
<tr>
<td>Avaya SIP Application Server PC</td>
<td>Windows XP Professional with Service Pack 2</td>
</tr>
</tbody>
</table>

**Table 1: Components and Versions**
3. Avaya Communication Manager Configuration

The basic configuration of Avaya Communication Manager and the endpoints that connect to or register with Avaya Communication Manager is not the focus of these Application Notes. This configuration information is provided in References [1-3]. For these Application Notes, it is assumed that SIP connectivity to Avaya SES, that all of the endpoints shown in Figure 1, and that PSTN connectivity have been configured.

The Conferencing application sets up and manages conferences, makes calls to endpoints, accepts calls from endpoints, and plays audio messages to endpoints. To support incoming calls from the endpoints into the Conferencing service, a SIP trunk was configured via Avaya SES. Additionally, UDE was configured as an adjunct system to Avaya SES (see Section 4.2). A hunt group and a group number were configured in Avaya Communication Manager to link the adjunct system on Avaya SES to Avaya Communication Manager. The group number, which is also referred to as the “dial-in number”, is the number that was dialed by the endpoints to reach UDE and to connect to the Conferencing service. To support outgoing calls from the Conferencing application to the endpoints, the same SIP trunk that was configured for outgoing calls, from Avaya Communication Manager to UDE, was used for incoming calls to Avaya Communication Manager from UDE.

These Application Notes assume that the following are already in place:

- SIP trunks between Avaya Communication Manager and Avaya SES have been configured (see Section 3.1.4.2).
- A PSTN trunk has been configured.
- All stations have been configured.
- Routing to the PSTN via Automatic Route Selection (ARS) has been configured and an appropriate route pattern has been configured.
- The Automatic Alternate Routing (AAR) code has been configured.
- Routing for calls to the Conferencing service dial-in number has been configured via AAR.

This section focuses on the configuration of the SIP trunks used for the calls between the UDE and Avaya Communication Manager and the configuration of the hunt group to allow endpoints to dial into the Conferencing service on UDE. For ease of configuration, the same network region and codec set was used by all of the components in the system. In this configuration, all components were configured to use IP network region “1” and IP codec set “1”. Media shuffling was enabled for all components within each site. Codec set “1” contains the “G.711MU” and “G.729AB codecs”.
To configure the hunt group and the SIP trunks, the following procedures were used:

- Configure SIP trunks
  - Display IP node names
  - Display IP codec set and IP network region
  - Configure SIP signaling group and trunk group
- Configure dial-in number to reach UDE
  - Display AAR feature access code
  - Configure hunt group
  - Configure route pattern to reach UDE
  - Configure AAR digit analysis table

The following configuration of Avaya Communication Manager was performed using the System Access Terminal (SAT). After completion of the configuration in this section, use the “save translation” command to make the changes permanent.

### 3.1. Configure SIP Trunks

#### 3.1.1. Display IP Node Names

Use the “display node-names ip” command to view the entries for the system processor and for Avaya SES. In this case, “procr” and “150.1.1.100” are listed under **Name** and **IP Address** for the system processor and “sesisv” and “150.1.1.150” are listed under **Name** and **IP Address** for Avaya SES. The actual node names and IP addresses may vary.

<table>
<thead>
<tr>
<th>Name</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>procr</td>
<td>150.1.1.100</td>
</tr>
<tr>
<td>sesisv</td>
<td>150.1.1.150</td>
</tr>
</tbody>
</table>

#### 3.1.2. Display IP Codec Set

In the sample configuration, codec set “1” was used for all components. Use the “display ip-codec-set n” command, where “n” is an existing codec set number. For this configuration, the G.711MU (µ-law) and G.729AB codecs are supported. The actual codecs may vary.

<table>
<thead>
<tr>
<th>Codec Set: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
</tr>
<tr>
<td>Codec</td>
</tr>
<tr>
<td>1: G.711MU</td>
</tr>
<tr>
<td>2: G.729AB</td>
</tr>
</tbody>
</table>
3.1.3. Display IP Network Region
In the sample configuration, network region “1” was used for all components. Use the “display ip-network-region n” command, where “n” is an existing network region number. Note that Authoritative Domain is set to “isvabc.com” which is the SIP domain for Avaya SES as shown in Section 4.1.

<table>
<thead>
<tr>
<th>display ip-network-region 1</th>
<th>Page 1 of 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP NETWORK REGION</td>
<td></td>
</tr>
<tr>
<td>Region: 1</td>
<td></td>
</tr>
<tr>
<td>Location: 1</td>
<td>Authoritative Domain: isvabc.com</td>
</tr>
<tr>
<td>Name: Network Region 1</td>
<td></td>
</tr>
<tr>
<td>MEDIA PARAMETERS</td>
<td></td>
</tr>
<tr>
<td>Intra-region IP-IP Direct Audio: yes</td>
<td></td>
</tr>
<tr>
<td>Codec Set: 1</td>
<td>Inter-region IP-IP Direct Audio: yes</td>
</tr>
<tr>
<td>UDP Port Min: 2048</td>
<td>IP Audio Hairpinning? n</td>
</tr>
<tr>
<td>UDP Port Max: 3329</td>
<td></td>
</tr>
</tbody>
</table>

3.1.4. Configure SIP Signaling Group and Trunk Group
In the sample configuration, trunk group “1” and signaling group “1” were used to reach SIP endpoints registered with Avaya SES. A separate trunk group “4” and a separate signaling group “4” were added to reach UDE.

3.1.4.1 Display SIP Trunk and Signaling Groups for Avaya SES Endpoints
Use the “display trunk-group n” command, where “n” is an available trunk group number. As shown below, trunk group “1” uses signaling group “1”.

<table>
<thead>
<tr>
<th>display trunk-group 1</th>
<th>Page 1 of 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUNK GROUP</td>
<td></td>
</tr>
<tr>
<td>Group Number: 1</td>
<td>Group Type: sip</td>
</tr>
<tr>
<td>Group Name: Avaya SES isvabc.com</td>
<td>CDR Reports: y</td>
</tr>
<tr>
<td>COR: 1</td>
<td>Cor: 1</td>
</tr>
<tr>
<td>TN: 1</td>
<td>TAC: #01</td>
</tr>
<tr>
<td>Direction: two-way</td>
<td>Outgoing Display? n</td>
</tr>
<tr>
<td>Dial Access? n</td>
<td>Night Service:</td>
</tr>
<tr>
<td>Queue Length: 0</td>
<td></td>
</tr>
<tr>
<td>Service Type: tie</td>
<td>Auth Code? n</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Signaling Group: 1</th>
<th>Number of Members: 20</th>
</tr>
</thead>
</table>
Use the “display signaling-group n” command, where “n” is an available signaling group number. Note that Far-end Domain is set to “isvabc.com” which is the SIP domain for Avaya SES as shown in Section 4.1.

```
display signaling-group 1
SIGNALING GROUP
Group Number: 1          Group Type: sip
Transport Method: tls

Near-end Node Name: procr               Far-end Node Name: sesisv
Near-end Listen Port: 5061               Far-end Listen Port: 5061
Far-end Network Region: 1

Far-end Domain: isvabc.com
Bypass If IP Threshold Exceeded? n

DTMF over IP: rtp-payload              Direct IP-IP Audio Connections? y
IP Audio Hairpinning? n
Enable Layer 3 Test? n
Session Establishment Timer(min): 3
```

3.1.4.2 Configure SIP Trunk and Signaling Groups to Connect to UDE

In the sample configuration, trunk group “4” and signaling group “4” were used to connect to UDE. Use the “add signaling-group n” command, where “n” is an available signaling group number. Enter the following values for the specified fields and retain the default values for all remaining fields. Submit these changes.

- **Group Type:** “sip”
- **Transport Method:** “tls”
- **Near-end Node Name:** System processor node name from Section 3.1.1
- **Far-end Node Name:** Avaya SES node name from Section 3.1.1
- **Near-end Listen Port:** “5061”
- **Far-end Listen Port:** “5061”
- **Far-end Network Region:** Avaya network region number from Section 3.1.3
- **Far-end Domain:** SIP domain name of UDE from Section 5.1

```
add signaling-group 4
SIGNALING GROUP
Group Number: y          Group Type: sip
Transport Method: tls

Near-end Node Name: procr               Far-end Node Name: sesisv
Near-end Listen Port: 5061               Far-end Listen Port: 5061
Far-end Network Region: 1

Far-end Domain: isvubi.com
Bypass If IP Threshold Exceeded? n

DTMF over IP: rtp-payload              Direct IP-IP Audio Connections? y
IP Audio Hairpinning? n
Enable Layer 3 Test? n
Session Establishment Timer(min): 3
```
Use the “add trunk-group n” command, where “n” is an available trunk group number. Enter the following values for the specified fields and retain the default values for the remaining fields. Submit these changes.

- **Group Type:** “sip”
- **Group Name:** A descriptive name
- **TAC:** An available trunk access code consistent with the dial plan
- **Service Type:** “tie”
- **Signaling Group:** Signaling group number for newly created signaling group
- **Number of Members:** Desired number of trunk group members

```
add trunk-group 4
```

<table>
<thead>
<tr>
<th>TRUNK GROUP</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Number: 6</td>
<td></td>
</tr>
<tr>
<td>Group Type: sip</td>
<td></td>
</tr>
<tr>
<td>CDR Reports: y</td>
<td></td>
</tr>
<tr>
<td>Group Name: UDE isvubi.com</td>
<td></td>
</tr>
<tr>
<td>Direction: two-way</td>
<td></td>
</tr>
<tr>
<td>Dial Access? n</td>
<td></td>
</tr>
<tr>
<td>Queue Length: 0</td>
<td></td>
</tr>
<tr>
<td>Service Type: tie</td>
<td>Auth Code? n</td>
</tr>
<tr>
<td>Signaling Group: 4</td>
<td></td>
</tr>
<tr>
<td>Number of Members: 10</td>
<td></td>
</tr>
</tbody>
</table>

3.2. Configure Dial-in Number to Reach UDE

This section focuses on the configuration necessary to allow endpoints to connect to the Conferencing service running on UDE.

### 3.2.1. Display AAR Feature Access Code

Use the “display feature-access-codes” command. As shown below, **Auto Alternate Routing (AAR) Access Code** is set “8”. This value will be used in Section 3.2.2 for the configuration of the hunt group.

```
display feature-access-codes
```

<table>
<thead>
<tr>
<th>FEATURE ACCESS CODE (FAC)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Abbreviated Dialing List1 Access Code:</td>
<td></td>
</tr>
<tr>
<td>Abbreviated Dialing List2 Access Code:</td>
<td></td>
</tr>
<tr>
<td>Abbreviated Dialing List3 Access Code:</td>
<td></td>
</tr>
<tr>
<td>Abbreviated Dial - Prgm Group List Access Code:</td>
<td></td>
</tr>
<tr>
<td>Announcement Access Code:</td>
<td></td>
</tr>
<tr>
<td>Answer Back Access Code:</td>
<td></td>
</tr>
<tr>
<td>Attendant Access Code:</td>
<td></td>
</tr>
<tr>
<td><strong>Auto Alternate Routing (AAR) Access Code: 8</strong></td>
<td></td>
</tr>
<tr>
<td>Auto Route Selection (ARS) - Access Code 1: 9</td>
<td>Access Code 2:</td>
</tr>
</tbody>
</table>
3.2.2. Configure Hunt Group

Configure a hunt group to define the extension, also referred to as the “dial-in number”, that will be used for Avaya Communication Manager endpoints to dial into the Conferencing service. Hunt groups are regularly used for interaction with voice mail systems. In this case, the hunt group works to connect Avaya Communication Manager endpoints to the Conferencing service that can be configured on UDE.

Use the “add hunt-group n” command, where “n” is a hunt group number. Enter the following values for the specified fields and retain the default values for the remaining fields.

- **Group Name:** A descriptive name
- **Group Extension:** An available number that is within the dial plan
- **ISDN/SIP Caller Display:** “mbr-name”

<table>
<thead>
<tr>
<th>add hunt-group 1</th>
<th>Page 1 of 60</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Number: 1</td>
<td>ACD? n</td>
</tr>
<tr>
<td>Group Name: Ubiquity</td>
<td>Queue? n</td>
</tr>
<tr>
<td>Group Extension: 27000</td>
<td>Vector? n</td>
</tr>
<tr>
<td>Group Type: ucd-mia</td>
<td>Coverage Path:</td>
</tr>
<tr>
<td>TN: 1 Night Service Destination:</td>
<td>MM Early Answer? n</td>
</tr>
<tr>
<td>COR: 1</td>
<td>Security Code:</td>
</tr>
<tr>
<td>ISDN/SIP Caller Display: mbr-name</td>
<td></td>
</tr>
</tbody>
</table>

On Page 2, enter the following values for the specified fields. Submit these changes.

- **Message Center:** “sip-adjunct”
- **Voice Mail Number:** Same number entered for **Group Extension** on Page 1
- **Voice Mail Handle:** A name for the UDE adjunct system that will be used in Avaya SES (see Section 4.2)
- **Routing Digits:** AAR access code defined in **Section 3.2.1**

<table>
<thead>
<tr>
<th>Message Center: sip-adjunct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voice Mail Number</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>27000</td>
</tr>
</tbody>
</table>
3.2.3. Configure Route Pattern to Reach UDE

Create a route pattern that will be used to route the call to the UDE “dial-in number” configured in Section 3.2.2.

Use the “change route-pattern n” command, where “n” is an available route pattern number. Enter the following values for the specified fields and retain the default values for the remaining fields.

- **Pattern Name:** A descriptive name
- **Grp No:** The trunk group number from Section 3.1.4.2
- **FRL:** Enter a level that allows access to this trunk, with 0 being least restrictive

```
change route-pattern 4

Pattern Number: 4 Pattern Name: To isvubi.com

Grp FRL NPA Pfx Hop Toll No. Inserted DCS/ IXC
No Mrk Lmt List Del Digits QSIG

1: 4 0 n user
2: n user
3: n user
4: n user
5: n user
6: n user

BCC VALUE TSC CA-TSC ITC BCIE Service/Feature PARM No. Numbering LAR
0 1 2 M 4 W Request Dgts Format Subaddress
1: y y y y y n n rest none
2: y y y y y n n rest none
```

3.2.4. Configure AAR Digit Analysis Table

Create an entry in the AAR Digit Analysis Table that will direct calls to the UDE “dial-in number” to the route pattern created in Section 3.2.3. In this configuration, the entire number is entered. A partial match, such as “270” or “27” would have worked also.

Use the “change aar analysis 0” command and add an entry to specify how calls to “27000” are routed. Enter the following values for the specified fields and retain the default values for the remaining fields. Submit these changes.

- **Dialed String:** Dialed prefix digits to match on, in this case “27000”
- **Total Min:** Minimum number of digits
- **Total Max:** Maximum number of digits
- **Route Pattern:** The route pattern number from Section 3.2.3
- **Call Type:** “aar”

```
change aar analysis 0

AAR DIGIT ANALYSIS TABLE

Percent Full: 1

Dial String Total Min Total Max Route Pattern Call Type Node ANI Num Req
27000 5 5 4 aar n n
```
4. Avaya SIP Enablement Services Configuration

This section provides the procedures for configuring SIP trunking between Avaya Communication Manager and Avaya SES and for configuring an adjunct system for connecting to UDE. Other standard configuration information for Avaya SES is covered in Reference [1]. It is assumed that Avaya SES is already installed and configured for connectivity with Avaya Communication Manager. It is assumed that all SIP endpoints are already administered. It is also assumed that media server address maps are already administered to route calls to non-SIP endpoints and to route calls to the PSTN.

In this sample configuration, calls made to the UDE “dial-in number”, “27000” in this configuration, were forwarded to UDE as configured in the Avaya SES adjunct system (see Section 4.2). On reaching UDE, these calls were answered by the Conferencing service and then acted upon by the Conferencing sample application.

The procedures covered in this section include the following areas:

- Display SIP domain
- Configure adjunct system
- Configure adjunct server

The configuration of Avaya SES was performed using the administration web interface. Access to the administration web interface is available by using the URL “http://<ip-address>/admin” in an Internet browser window, where “<ip-address>” is the IP address of Avaya SES. Note that the IP address for Avaya SES may vary. In the sample configuration, “150.1.1.150” was used to access Avaya SES. Log in with the appropriate credentials and click the **Launch Administration Web Interface** link.
The **Top** screen is displayed, as shown below. These Application Notes assume Avaya SES has already been configured with the proper domain and host information.
4.1. Display SIP Domain
Select Server Configuration → System Properties from the left pane to display the View System Properties screen. Use the value in the SIP Domain field (in this case “isvabc.com”) for configuring Authoritative Domain in Section 3.1.3 and for configuring Far-end Domain in Section 3.1.4.2.
4.2. Configure Adjunct System

Administer Avaya SES to add UDE as an adjunct system. From the Top screen, select Adjunct Systems → List from the left pane to display the List Adjunct Systems screen. Click the Add Another Adjunct System link in the right pane.

The Add Adjunct screen is displayed next. For the System Name field, enter the value entered for Voice Mail Handle in the Hunt Group form in Section 3.2.2 (in this case, “udeubi”). Click Add.

Click Continue on the Continue screen that confirms that the adjunct system was added (not shown). An updated List Adjunct Systems screen is displayed showing the newly created adjunct system (see Section 4.3).
NOTE: By adding UDE as an adjunct system, a “trusted host” entry is automatically created for UDE. That is, UDE becomes trusted to Avaya SES and Avaya SES will not challenge calls initiated from UDE to the Avaya endpoints.

4.3. Configure Adjunct Server

Administer an adjunct server for the newly created adjunct system to associate UDE with the Group Extension that was created in Avaya Communication Manager in Section 3.2.2. Continuing from Section 4.2, click the List Adjunct Servers(0) link in the right pane of the List Adjunct Systems screen.

![List Adjunct Systems Screen](image)

The List Adjunct Servers screen is displayed next. Click the Add Another Adjunct Server to System udeubi link in the right pane.

![Add Another Adjunct Server Screen](image)
The **Add Adjunct Server** screen is displayed next. Enter the following values for the specified fields. Click **OK**.

- **Server Name:** A descriptive name that is different from the adjunct system name, such as “udeubi-1”
- **Server ID:** A number that is different from the **Group Extension** defined in **Section 3.2.2** and that falls within the administered dial plan in Avaya Communication Manager.
- **Link Type:** “TCP”
- **Server IP Address:** The IP address of UDE, from **Section 5.1**.

![Add Adjunct Server Screen](image-url)
Click **Continue** on the **Continue** screen that confirms that the adjunct server was added to the adjunct system (not shown). An updated **List Adjunct Servers** screen is displayed. To save all of the changes made in this section, click the **Update** link in the bottom left pane.
5. Ubiquity Developer Edition Configuration

The installation of UDE is covered in Reference [5]. For this sample configuration, UDE is installed on a system running the Microsoft Windows operating system. The components listed below are assumed to be installed. The installation and configuration of these components are covered in Reference [5].

- Sun Microsystems Java SDK 5.0
- Eclipse SDK 3.2 and GEF
- Ubiquity Developer Edition 1.3
- SIP Service Oriented Architecture 1.1
- Ubiquity Development Studio 1.3
- Service Components Start-up Pack 1.5

This section focuses on the procedures to configure UDE to support the Conferencing service and sample application. The procedures are as follows:

- Display UDE configuration
- Deploy extension and services
  - Deploy Distributed State Management (DSM)
  - Deploy Service Components (service_components.sar)
  - Deploy Conferencing services (conferencing.sar)
  - Deploy Conferencing sample application (conferencing_sample_blb.sar)
- Configure SOOF aliases
  - Configure Service Components aliases
  - Configure Conferencing service aliases
  - Configure Conferencing sample application aliases
The configuration of UDE is done using the Eclipse SDK. Start the Eclipse SDK by opening the “eclipse.exe” executable file. For this sample configuration, the file is located at “C:\Ubiquity\Eclipse\eclipse.exe”. The location of the file will vary based on the installation paths chosen by the installer.

The **Ubiquity Developer Studio – Eclipse SDK** window is displayed. The area highlighted in red below is the **UDE View**. The **UDE View** is the portion of the **Ubiquity Developer Studio – Eclipse SDK** window that will be referenced in this section. **NOTE**: The appearance of the **Ubiquity Developer Studio – Eclipse SDK** window may vary as the window can be organized so that the sections within the window are shown differently.
5.1. Display UDE Configuration

Display the domain(s) controlled by UDE by selecting Window → Preferences → UDS Configuration → UDE Configuration from the Ubiquity Developer Studio – Eclipse SDK window. The Preferences window is displayed. Under Defined Profiles is a list of profiles configured for UDE. For this sample configuration, only one profile was configured. In this case, that profile is “Broadcom”. Select Broadcom and click Edit.
The **UDE Profile Configuration** window is displayed. Note the list shown under **Controlled Domains**. In this case, UDE is controlling “isvubi.com” and “150.1.1.170”. The IP address is used in the Avaya SES configuration in **Section 4.3**. Click **Cancel** and click **Cancel** again to return to the **Ubiquity Developer Studio – Eclipse SDK** window.

![UDE Profile Configuration](image)

### 5.2. Deploy Extension and Services

To support the Conferencing application, the following extension, services, and application must be deployed.

- **dsm_memory.jar**: This file is used for the DSM extension. The DSM extension is required for the SIP Service Components services.
- **service_components.sar**: This file includes a collection of SIP Service Components. Based on the recommendation from Reference [7], this file was used for this sample configuration. However, the Service Components can be deployed as individual services using individual files for each service.
- **conferencing.sar**: This file is for the Conferencing service.
- **conferencing_sample_blb.sar**: This file is for the Conferencing sample application.

The following procedures assume that the above services have not already been deployed.
5.2.1. Deploy Distributed State Management Extension

From the Ubiquity Development Studio – Eclipse SDK window shown in Section 5, select Window → Show View → Other. The Show View dialogue is displayed. Expand Ubiquity Developer Studio and click Extensions.
The Extensions tab is displayed along with the UDE View tab. Select the Service Host tab in the Extensions tab. The DSM extension is listed as dsm_memory.jar in the Undeployed Extensions frame. Right-click on dsm_memory.jar and click Deploy on the pop-up menu (not shown).

The Attribute Configuration :dsm_memory.jar dialogue is displayed. Retain the default values and click OK to close the dialogue and to deploy the extension.
The DSM extension (*dsm_memory.jar*) is now displayed in the **Deployed Extensions** frame.
5.2.2. Deploy Service Components (service_components.sar)

From the **UDE View** tab of the **Ubiquity Developer Studio – Eclipse SDK** window, scroll through the list under **Undeployed Services and Applications** and find the entry for **service_components.sar**. Right-click on **service_components.sar** and click **Deploy** on the pop-up menu (not shown).
The **Attribute Configuration : service_components.sar** dialogue is displayed. Enter “conferencing_SampleBLB” into the **Incoming Dialog Observer Alias** field. This is required per the instructions provided in the “sampleapp.bat” file described in **Section 6.1**. Click **Add** under **Media Server Details** to configure the media server.

![Attribute Configuration : service_components.sar](image)

The **Enter attribute st…** dialogue is displayed.

The configuration of the **service_components.sar** service includes the details for the Cantata SnowShore IP Media Server. A Convedia IP Media Server is also supported by UDE but the SnowShore IP Media Server was used in this sample configuration. Enter a string in the form of the following into the field:

```
"name=<name>,type=<ip-media-driver>,alias=<driver-alias>,ip=<ip-address>,ports=<#-ports>"
```

- **name**: A descriptive name for the IP media server being configured. More than one IP media server is supported. For this sample configuration, “ms1” was used.
- **type**: The type of IP media server being configured. For this sample configuration, “snowshore_g2” was entered for the SnowShore IP Media Server.
- **alias**: The alias used internally by the Service Components to refer to the IP media server driver. For this sample configuration, it is “snowshoredriver”.
- **ip**: The IP address of the IP media server. In this case, it is “150.1.1.140”.
- **ports**: Maximum number of ports supported by the IP media server. In this case, it is 50.
The string entered for this sample configuration was as follows:

“name=ms1,type= snowshore_g2,alias=snowshoredriver,ip=150.1.1.140,ports=50”

Click OK.

The Attribute Configuration : service_components.sar dialogue is displayed. Leave the other fields at their default values. Click OK to close the dialogue and to deploy the service.

Once deployed, service_components.sar is displayed under Deployed Services and Applications in the UDE View tab as shown in Section 5.2.3.
5.2.3. Deploy Conferencing Service (conferencing.sar)

Continuing from the deployment of the service_components.sar service, deploy the conferencing.sar service. From the UDE View tab of the Ubiquity Developer Studio – Eclipse SDK window, scroll through the list under Undeployed Services and Applications and find the entry for conferencing.sar. Right-click on conferencing.sar and click Deploy on the pop-up menu (not shown).

The Attribute Configuration : conferencing.sar dialogue is displayed (only a part of the dialogue is shown below).
The **Attribute Configuration : conferencing.sar** dialogue includes the fields shown in **Table 2**. The values for these fields are explained and defined in the “sampleapp.bat” file described in **Section 6.1**.

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application ID</td>
<td>“sampleapp”</td>
</tr>
<tr>
<td>Application BaseName</td>
<td>“sampleappbase”</td>
</tr>
<tr>
<td>Application PropertiesPath</td>
<td>“file:///C:/Ubiquity/USCL/ServiceComponents-1.5.2/samples/conferencing_services_sample”</td>
</tr>
<tr>
<td></td>
<td>▪ The path to access the sample application properties file</td>
</tr>
<tr>
<td></td>
<td>▪ The value above is based on the folder in which the sample application is stored: “C:\Ubiquity\USCL\ServiceComponents-1.5.2\samples\conferencing_services_sample”</td>
</tr>
<tr>
<td></td>
<td>▪ The path to access the messages from the Cantata SnowShore IP Media Server</td>
</tr>
<tr>
<td></td>
<td>▪ The path to access the conference recordings from the Cantata SnowShore IP Media Server</td>
</tr>
<tr>
<td></td>
<td>▪ The path to access the general recordings from the Cantata SnowShore IP Media Server</td>
</tr>
</tbody>
</table>

**Table 2: Conferencing Service SAR Fields**

All of the fields in the **Attribute Configuration : conferencing.sar** dialogue are configured in the same manner. The following shows the configuration steps for the **Application ID** field. Click **Add** for the **Application ID** field. The **Enter attribute st** dialogue is displayed. Enter “sampleapp” in the field, as listed for **ApplicationID** in **Table 2**. Click **OK**.
Repeat the above steps to add the remaining fields listed in Table 2. Click OK to close the dialogue and to deploy the service.

Once deployed, **conferencing.sar** is displayed under **Deployed Services and Applications** as shown in the **UDE View** tab as shown in **Section 5.2.4**.
5.2.4. Deploy Conferencing Sample Application
(conferencing_sample_blb.sar)

Continuing from the deployment of the conferencing.sar service, deploy the conferencing_sample_blb.sar application. From the UDE View tab of the Ubiquity Developer Studio – Eclipse SDK window, scroll through the list under Undeployed Services and Applications and find the entry for conferencing_sample_blb.sar. Right-click on conferencing_sample_blb.sar and click Deploy on the pop-up menu (not shown).

There are no configuration parameters for this file. Once deployed, conferencing_sample_blb.sar is displayed under Deployed Services and Applications as shown in Section 5.3.1.
5.3. Configure SOOF Aliases

Once the services and application listed in Section 5.2 have been deployed, SOOF aliases must be configured for the deployed services and application.

5.3.1. Configure Service Components Aliases

In the UDE View tab, click the Menu button (≡) to open the UDE View menu. Select Import Soof Configuration.
The **Soof Configuration Import** dialogue is displayed. Select **UEM** and click **Browse** for **File Location** to find the SOOF alias file. For this sample configuration, the **services_components.sar** service is used. Therefore, the alias file that must be used is the “single_sar_service_components.alias” file. This file is located in the “alias” sub-folder where the Service Components files were installed. In this sample configuration, the alias file is located in the “C:\Ubiquity\USCL\ServiceComponents-1.5.2\alias\” folder. If the Service Components were deployed as individual services, a different alias file would be used. Click **Finish** to complete the configuration of the SOOF aliases for the **service_components.sar** service.

![Soof Configuration Import Dialogue](image)

### 5.3.2. Configure Conferencing Service Aliases

As shown in **Section 5.3.1**, click the **Menu** button (✓), in the **UDE View** tab, to open the **UDE View** menu. Select **Import Soof Configuration**.

Select **UEM** and click **Browse** for **File Location** to find the SOOF alias file. The alias file is the “conferencing.alias” file. This file is located in the “alias” sub-folder where the Service Components files were installed. In this sample configuration, the alias file is located in the “C:\Ubiquity\USCL\ServiceComponents-1.5.2\alias\” folder. Click **Finish** to complete the configuration of the SOOF aliases for the **conferencing.sar** service.
5.3.3. Configure Conferencing Sample Application Aliases

To configure the aliases for the Conferencing sample application, open the SOOF Routing tab. From the Ubiquity Development Studio – Eclipse SDK window shown in Section 5, select Window → Show View → Other. The Show View dialogue is displayed. Expand Ubiquity Developer Studio and click SOOF Routing.

The SOOF Routing tab is displayed next to the UDE View tab. Select “conferencing_sample_blb” from the Deployed Components drop-down list and “conferencing_sample_blb/ConferencingSampleBlb” from the Servlets within the Component drop-down list. Click the Add a new Alias button at the top right of the SOOF Routing tab.
The **Add an Alias** dialogue is displayed. Enter “conferencing_SampleBLB” in the **Alias** field. Verify that “conferencing_sample_blb” is displayed for the **Component** drop-down list and that “conferencing_sample_blb/ConferencingSampleBlb” is displayed for the **Servlet** drop-down list. Click **OK**. The alias “conferencing_SampleBLB” is then displayed under **Aliases For the Servlet** in the **SOOF Routing** tab (not shown).

![Add an Alias dialogue]

From the **SOOF Routing** tab, verify that “conferencing_sample_blb” is selected in the **Deployed Components** drop-down list and select “conferencing_sample_blb/ConferencingSampleBlbIncomingDialogListener” from the **Servlets within the Component** drop-down list. Click the **Add a new Alias** button at the top right of the **SOOF Routing** tab.

![SOOF Routing tab]
The Add an Alias dialogue is displayed. Enter “conferencing_SampleBLBIncomingDialogListener” in the Alias field. Verify that “conferencing_sample_blb” is displayed for the Component field and that “conferencing_sample_blb/ConferencingSampleBlbIncomingDialogListener” is displayed for the Servlet field. Click OK. The alias “conferencing_SampleBLBIncomingDialogListener” is then displayed under Aliases For the Servlet in the SOOF Routing tab (not shown).

6. Conferencing Sample Application Configuration

The configuration of the Conferencing sample application is defined in the “sampleapp.bat” file that is installed in the sample application folder (see Table 2 in Section 5.2.3). Additional information on using this file and the sample application is provided in Reference [8]. As this sample configuration is using a Windows operating system, the batch file (“.bat”) version of the file is used. The “sampleapp.sh” file would be used in a Linux installation.

The procedures covered in this section include the following areas:

- Configure Conferencing sample application start-up file
- Copy dependent Java archive files

6.1. Configure Conferencing Sample Application Start-Up File

The “sampleapp.bat” file (see Table 2 in Section 5.2.3) includes instructions to configure the Conferencing sample application. The file also holds configuration parameters for the sample application. The description of these parameters is included in this file. For the DOMAIN_NAME field, enter the domain name or IP address of UDE. In this configuration, the domain name is “isvubi.com”. Retain the default values for the remaining fields or change them to match the configuration. Save the file. The following is an excerpt of the configuration portion of the file. For this configuration, the defaults were used for the remaining fields.

```
set CLASSPATH="conferencing_sample_app.jar;sassdk.jar;soof_api.jar;service_components_api.jar"
set APP_NAME="sampleapp"
set DOMAIN_NAME="isvubi.com"
set NSREG_PORT="1098"
set CONFERENCING_BLB_NAME="conferencing_sample_blb"
set LOAD_FILE_NAME=""
```
6.2. Copy Dependent Java Archive Files

Per the instructions in the “sampleapp.bat” file (see Table 2 in Section 5.2.3), the following Java archive (JAR) files must be copied to the sample application folder:

- conferencing_sample_app.jar
- sassdk.jar;soof_api.jar
- service_components_api.jar

The location of the above files is provided in the “sampleapp.bat” file.

7. Verification Steps

This section provides the tests that can be performed on Avaya Communication Manager, Avaya SES, and on the Eclipse SDK to verify proper configuration of Avaya Communication Manager, Avaya SES, and UDE.

7.1. Verify Avaya Communication Manager

7.1.1. Verify Idle Trunk and Signaling Groups

With an idle system (or at least, with no calls active on the SIP trunk to reach UDE), verify the status of the SIP trunk group used to reach UDE by using the “status trunk n” command, where “n” is the trunk group number administered in Section 3.1.4.2. Verify that all trunk group members are in the “in-service/idle” state as shown below.

```
status trunk 4
```

```
<table>
<thead>
<tr>
<th>Member</th>
<th>Port</th>
<th>Service State</th>
<th>Mtce Connected Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>0004/001</td>
<td>T00111</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/002</td>
<td>T00112</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/003</td>
<td>T00113</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/004</td>
<td>T00114</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/005</td>
<td>T00115</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/006</td>
<td>T00116</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
</tbody>
</table>
```

Verify the status of the SIP signaling group associated with the above trunk group by using the “status signaling-group n” command, where “n” is the signaling group number administered in Section 3.1.4.2. Verify that the signaling group is “in-service” as indicated in the Group State field shown below.

```
status signaling-group 4
```

```
<table>
<thead>
<tr>
<th>Group ID: 4</th>
<th>Active NCA-TSC Count: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Type: sip</td>
<td>Active CA-TSC Count: 0</td>
</tr>
<tr>
<td>Signaling Type: facility associated signaling</td>
<td>Group State: in-service</td>
</tr>
</tbody>
</table>
```

Verify the status of the SIP signaling group associated with the above trunk group by using the “status signaling-group n” command, where “n” is the signaling group number administered in Section 3.1.4.2. Verify that the signaling group is “in-service” as indicated in the Group State field shown below.
7.1.2. Verify Active Trunk Group

Prior to executing the Conferencing sample application, make a call to the pilot number configured to reach UDE; in this configuration, “27000”. Verify that the call was made over the proper trunk group. Verify the status of the SIP trunk group used to reach UDE by using the “status trunk n” command, where “n” is the trunk group number administered in Section 3.1.4.2. Verify that one trunk group member is in the “in-service/active” state as shown below.

<table>
<thead>
<tr>
<th>Member</th>
<th>Port</th>
<th>Service State</th>
<th>Mtce Connected Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>0004/001</td>
<td>T00111</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/002</td>
<td>T00112</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/003</td>
<td>T00113</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/004</td>
<td>T00114</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/005</td>
<td>T00115</td>
<td>in-service/idle</td>
<td>no</td>
</tr>
<tr>
<td>0004/006</td>
<td>T00116</td>
<td>in-service/active</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T00001</td>
</tr>
</tbody>
</table>

Verify the status of active trunk group member by using the command “status trunk x/y”, where “x” is the number of the SIP trunk group and “y” is the member number of a connected trunk. Verify that the Service State is “in-service/active”. In this scenario, a call was made but audio has not been established. Therefore, Audio Connection Type displays “ip-idle”.

<table>
<thead>
<tr>
<th>Trunk Group/Member: 0004/006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port: T00116</td>
</tr>
<tr>
<td>Service State: in-service/active</td>
</tr>
<tr>
<td>Maintenance Busy? no</td>
</tr>
<tr>
<td>Connected Ports: T00001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port</th>
<th>Near-end IP Addr : Port</th>
<th>Far-end IP Addr : Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>01A0017 150. 1. 1.100 : 5061</td>
<td>150. 1. 1.150 : 5061</td>
</tr>
</tbody>
</table>

Audio: 
Video: 
Video Codec: 
Authentication Type: None

Audio Connection Type: ip-idle
7.2. Verify Eclipse SDK
Verify that the SOOF aliases have been administered properly for the deployed services. Access the SOOF Routing tab using the instructions from Section 5.3.3. Select “**” from the Deployed Components drop-down list and “*” from the Servlets within the Component drop-down list. Scroll down in the Aliases For the Servlet list and verify that aliases similar to what is shown below are displayed. All of the aliases that were imported in Section 5.3 should be listed. This list in the SOOF Routing tab can be compared to the list found on opening the “single_sar_components.alias” text file (see Section 5.3.1), in the “conferencing.alias” text file (see Section 5.3.2), and in the “sampleapp.bat” text file (see Section 5.3.3).
7.3. Verify UDE
Verify that UDE has been configured properly to support the Conferencing sample application.

7.3.1. Verify “Dial-in” Calls into the Conferencing Application
Verify that calls made to “dial-in number” for UDE are received by the Conferencing sample application. Start the application by browsing to the location of the “sampleapp.bat” file (see Table 2, Section 5.2.3) and double-click on the file. The Ubiquity Conferencing Service Sample Application window is displayed.

Make a call to the “dial-in number” (“27000” for this configuration) and verify that an entry for this calling endpoint is displayed in the Dial-in Participant area of the Ubiquity Conferencing Service Sample Application window. In this example, an entry is displayed for extension “26005” with station name (Caller ID Name) of “Station 26005” and SIP URI of “sip:26005@isvabc.com”.

![Ubiquity Conferencing Service Sample Application](image-url)
7.3.2. Verify “Dial-out” Calls from the Conferencing Application

Verify that calls can be made from the Conferencing sample application. Start the application by browsing to the location of the “sampleapp.bat” file (see Table 2, Section 5.2.3) and double-click on the file. The Ubiquity Conferencing Service Sample Application window shown in Section 7.3.1 is displayed.

Enter a descriptive label into the Dial-out Participant field such as “26005” to represent endpoint station “26005”. Click Add. The Add Participant dialogue is displayed. Enter the SIP URI to reach endpoint “26005” into the To field. In this example, “sip:26005@isvabc.com” was entered. Retain the default values for the remaining fields. Click OK.
Verify that an entry for endpoint “26005” is displayed in the **Dial-out Participant** area of the **Ubiquity Conferencing Service Sample Application** window. Additionally, verify that there is a message in the **Messages** area showing that the endpoint was added.
Select the entry for endpoint “26005” in the **Dial-out Participant** area and drag the entry into the **IVR** area. Verify that a call is made to the endpoint. Answer the call. Verify that an entry is displayed for this endpoint in the **IVR** area. Additionally, verify that there is a message in the **Messages** area showing that the endpoint was added to the IVR session. Verify that the Caller ID Name shown on the endpoint is “SampleApplication”. This name is internally defined in the sample application and is sent in the SIP “From” header.
7.3.3. Verify Creation of a Conference Call

Verify that a conference can be created using the Conferencing sample application. Continue from Section 7.3.2 with the endpoint added to the IVR session and add the endpoint to a conference.

First a conference must be created. Enter a descriptive label into the Conferences field and click Create. The Create Conference dialogue is displayed. Retain the default values for all fields and click OK.
Verify that the newly created conference is displayed in the Conferences area under All Conferences. Verify that there are messages in the Messages area indicating that the conference was created successfully.
Right-click on the newly created conference and click **Start** from the pop-up menu.
Verify that there are messages in the **Messages** area indicating that the conference was started successfully. Select the entry for endpoint “26005” listed in the **IVR** area and drag it on top of the new conference. Verify that the entry for endpoint “26005” has been removed from the **IVR** area and has been moved under the created conference in the **Conferences** area. Verify that there are messages in the **Messages** area indicating that endpoint “26005” has been added to the conference.
7.4. Verification Scenarios
When the Conferencing service was deployed on UDE and the Conferencing sample application was executed to provide conferencing and IVR services, the following scenarios were verified.

- Digital, H.323, SIP, and PSTN endpoints can dial into the service hosted by the sample application and can be placed into an IVR session using the G.711 µ-law and G.729 codecs, with shuffling enabled and disabled. Once in the IVR session, an announcement can be played to the endpoint.
- Digital, H.323, SIP, and PSTN endpoints can dial into the service hosted by the sample application and can be placed into a conference using the G.711 µ-law and G.729 codecs, with shuffling enabled and disabled. Once in the conference, an announcement can be played to all of the endpoints placed into the conference.
- Digital, H.323, SIP, and PSTN endpoints can be called by the application and placed into an IVR session using the G.711 µ-law and G.729 codecs, with shuffling enabled and disabled. Once in the IVR session, an announcement can be played to the endpoint.
- Digital, H.323, SIP, and PSTN endpoints can be called by the application and placed into a conference using the G.711 µ-law and G.729 codecs, with shuffling enabled and disabled. Once in the conference, an announcement can be played to all of the endpoints placed into the conference.
- Proper display of the calling party information was verified for calls made to Digital, H.323, SIP, and PSTN endpoints.

8. Conclusion
As illustrated in these Application Notes, the Conferencing sample application for the Service Components hosted by Ubiquity Developer Edition can interoperate with Avaya Communication Manager using SIP trunks via Avaya SES. The sample application can be used for IVR sessions and for conferences. Endpoints can call into the service hosted by the sample application or can be called using the application.
9. Additional References

This section references the product documentation relevant to these Application Notes.


©2008 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 Solution & Interoperability Test Lab at interoplabnotes@list.avaya.com.