

Callingpolicies Sample Snap-in

Introduction

The Callingpolicies sample snap-in prompts the calling party to press a digit that will determine which one of the following operations is performed:

1. The call is forked to the configured destinations.
2. The call is redirected to a configured destination.
3. The call is allowed to proceed to its original destination (called party).
4. The call is dropped.

Forking operation adds participants to the call. A maximum of five participants can be added to the call using the forking operation. The fork destinations along with the called party will ring in this case. Redirect, diverts a call to a different destination. The Callingpolicies snap-in demonstrates forking operation of up to two participants. In addition, it also demonstrates the redirect, allow and drop operations.

Overview

The Callingpolicies snap-in is a term side, called party snap-in. This means that it is invoked for calls to users who have the snap-in enabled in their Service Profile. It also means that the snap-in is not invoked for users making calls.

When invoked, the sample snap-in instructs the Avaya Media Server (AMS) to play the Callingpolicies.wav file. The Callingpolicies.wav file is located at sample_services/Callingpolicies/Callingpolicies-war/src/main/webapp/Callingpolicies.wav in the SDK. The wav file is part of the sample snap-in war. This file does not need to be installed on Avaya Media Server. As long as Avaya Media Server is installed and configured properly according to the [\[DEV: Deployment and Administration guides\]](#), Avaya Media Server will be able to retrieve the wav file via HTTPs from the sample snap-in and play it to the caller.

Additionally, to establish trust between Avaya Media Server and Collaboration Environment for HTTPs to work, please refer to two sections of the “Administering Avaya Media Server 7.6” document, chapter4:

- “network settings configuration” -> “configuring connection security options”
- “importing a trust certificate to the trust store” -> “security configuration”

For assistance on exporting trusted certificate, please refer to “Administering Avaya Aura® Session Manager-release 6.3” in the “certificate management” section.

The snap-in waits for the caller to press a digit, and then performs the selected operation. If no input is received from the caller within 60 seconds, the call is dropped.

If an invalid digit, not corresponding to one of the four operations 1,2,3,4, is pressed, then the call is dropped as well.

Figure i, shows where the three attributes are administered. The first two ones represent the two fork destinations. The last one represents the redirect destination. The destinations can be any phone number. The forking operation proceeds to call both the configured destinations as well as the called party.

Based on the destination provided in the attributes page, the call is diverted to this destination.

Concepts Demonstrated

- Reading attributes from the user's Service Profile.
- Forking a call using the Collaboration API's `Call` method `addParticipant`. This method adds additional participants to the call.
- Redirecting a call using the Collaboration API's `Call` method `divertTo`. This method diverts a call to a different destination.
- Allowing the call to proceed to the called party using Collaboration API's `Call` method `allow`.
- Allowing the call to be dropped by the called party using Collaboration API's `Call` method `drop`.
- Generating and configuring a `playItem` object for the purposes of playing an announcement. This shows the use of methods such as `setSource`, used to set the wav file path, `setInterruptible`, that defines whether or not audio announcements may be interrupted when a digit is detected, and `setIterateCount`, that sets the number of times the prompt is played .
- Playing a prompt and collecting a digit.

Snap-in Attributes

The Callingpolicies snap-in defines the forking and redirect destinations as service attributes. The attributes are pre-set, prior to invoking the call, on the attributes page as follows:

The screenshot shows a configuration window for 'Service Profiles'. The 'Profile' dropdown is set to 'MyServiceProfile-01' and the 'Service' dropdown is set to 'CallingPolicies'. Below these, there is a table with 3 items. The table has columns for 'Name', 'Override Default', 'Effective Value', and 'Description'. The 'Override Default' column contains checkboxes, all of which are checked. The 'Effective Value' column contains text input fields with the values '3404@avaya.com', '3403@avaya.com', and '3402'. The 'Description' column contains the text 'Registered extension with domain' for all three rows. At the bottom right of the window, there are 'Commit' and 'Cancel' buttons.

Name	Override Default	Effective Value	Description
First fork destination	<input checked="" type="checkbox"/>	3404@avaya.com	Registered extension with domain
Next fork destination	<input checked="" type="checkbox"/>	3403@avaya.com	Registered extension with domain
Redirect destination	<input checked="" type="checkbox"/>	3402	Registered extension with domain

Figure i

Attributes are defined in the `properties.xml` descriptor.

Note that this example shows configuration of attributes at a Service Profile level. For more information on snap-in attributes, including information about configuring attributes at the cluster or global level, refer to the Service Development guide [\[DEV: Service Development Guide\]](#).

Detailed description

The service framework invokes the class `CallingPoliciesCallListener` recognized by the fact that it extends the `CallListenerAbstract` class, and is annotated with `TheCallListener`.

The `Call` Object is then passed on to the method `executePromptAndCollectOperation()` of the class `PromptAndCollectOperationImpl` that plays the announcement and collects a digit.

The above mentioned class makes use of `MediaService.createplayItem()` that sets the configuration of the prompt such as `setSource()` method, that specifies the wav file location, `setInterruptible()`, that defines whether or not audio announcements may be interrupted when a digit is detected, and `setIterateCount()`, that specifies the number of times prompt can be played.

In addition to creating `PlayItem`, `MediaService.digitOptions()` is used to specify digit options such as `setNumberOfDigits()`, that specifies the number of digits to be collected, `setTerminationKey()`, to set the key that would terminate the digit collection, `setTimeout()`, to specify the milliseconds to wait to receive the first digit. `MediaService.promptAndCollect()` is invoked to play the announcement to the participant.

Next, the `digitCollected()` method of the `CallingPoliciesMediaListener` class is

invoked, after the digit has been collected. In addition to validating the digit collected, it invokes the `forkTo()` method of the `ForkingOperationImpl` class and the `divertTo()`, `allow()` and `drop()` methods belonging to the `Call Collaboration API`. The `ServiceAttributeReaderImpl` class and `getForkedDestinations()` methods are used to obtain the service attributes from the attribute page.

Snap-in Invocation

There are two places that properties files have been configured to ensure that the Callingpolicies snap-in gets invoked appropriately.

In the `properties.xml` document under `Callingpolicies-svar/src/main/resources`, the following two properties have been populated, indicating that this is a called party snap-in:

```
<term_order>1</term_order>
<term_group>1</term_group>
```

Additionally, the following rule was populated in the `CARRule.xml` file located under `Callingpolicies-svar/src/main/resources`. This rule is required for any snap-in that uses Avaya Media Server, as Callingpolicies does to play announcements.

```
<TerminatingServiceRule desc="Interested in Callingpolicies featureURI as named
app">
  <FeatureURI>Callingpolicies</FeatureURI>
</TerminatingServiceRule>
```

Installation and Configuration

The Callingpolicies snap-in can be found in the SDK at:

`/SDK-3.0-Zip/samples/Callingpolicies`.

Change directory to where the snap-in resides and compile it (`mvn clean install`). Then load and install the svar on System Manager, enable it in Service Profiles, and administer the forking and redirect destinations for a Service Profile.

For information on installing the snap-in, assigning it to users, and configuring the display string attribute for Service Profiles see the [\[DEV: Installing, Configuring and Testing a CE Service guide\]](#).

Testing the Snap-in using an example

Assuming five users: called party: `3401@avaya.com`, calling party: `3402@avaya.com`, first fork destination: `3403@avaya.com`, next fork destination: `3404@avaya.com`, redirect destination: `3300@avaya.com`

1. Please note: the communication profile for the station form for the called user must be configured as shown in Figure ii below with the service profile included.
2. Call 3401 from an external number. 3401 is the called party.

3. The announcement is played.
4. Based on the digit pressed, call is forked, redirected, allowed to continue or dropped.
 - For 1= Fork, both fork destinations and the original destination are alerted. Please refer to the screen shot below – Figure ii.
 - For 2= redirect, it redirects to the destination provided in the service attribute. Please refer to the screen shot below – Figure ii.
 - For 3= allow, the called party (in our case, 3401@avaya.com) rings.
 - For 4= drop, the call is dropped and the calling party may hear a tone indicating that the call has been dropped.
 - For any other digit than 1,2,3,4, the call is dropped.

*Please Note: The caller cannot be a Communication Manager station.

The screenshot displays a configuration page with several sections:

- Application Sequences:** Includes dropdowns for 'Origination Sequence' and 'Termination Sequence', both currently set to '(None)'.
- Call Routing Settings:** Includes a dropdown for 'Home Location' set to 'CE development / Westminster' and a dropdown for 'Conference Factory Set' set to '(None)'.
- Call History Settings:** Includes a checkbox for 'Enable Centralized Call History?' which is checked.
- Collaboration Environment Profile:** This section is highlighted with a red box. It contains a checked checkbox and a dropdown menu for 'Service Profile' set to 'MyServiceProfile-01'.
- Other Profiles:** Below the highlighted section, there are several unchecked checkboxes for other profiles: 'CM Endpoint Profile', 'CS 1000 Endpoint Profile', 'Messaging Profile', 'CallPilot Messaging Profile', 'IP Office Endpoint Profile', 'Presence Profile', and 'Conferencing Profile'.

Figure ii

Troubleshooting

Problem:

When a call is made, no announcement is played.

Action:

1. Please make sure Avaya Media Server has been configured and is up and running.
2. Please make sure the snap-in has been assigned to the called party as shown in Figure ii.