Avaya Solution & Interoperability Test Lab

Application Notes for Jacada Visual IVR with Avaya Aura® Orchestration Designer and Avaya Aura® Experience Portal R6.0 – Issue 1.0

Abstract

These Application Notes describe the configuration steps required to integrate Jacada Visual IVR with Avaya Aura® Orchestration Designer and Avaya Aura® Experience Portal. Jacada Visual IVR provides the user an alternative method to access Interactive Voice Response (IVR) applications created using Avaya Aura® Orchestration Designer. Traditionally IVR applications are accessed by voice or telephone keypad strokes via PSTN lines. With Jacada Visual IVR the user can access the same IVR applications via a web browser based interface on a computer or a mobile device.

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 configuration steps required to integrate Jacada Visual IVR with Avaya Aura® Orchestration Designer and Avaya Aura® Experience Portal. Jacada Visual IVR provides the user an alternative method to access Interactive Voice Response (IVR) applications created using Avaya Aura® Orchestration Designer. Traditionally IVR applications are accessed by voice or telephone keypad strokes via PSTN lines. With Jacada Visual IVR the user can access the same IVR applications via a web browser based interface on a computer or a mobile device.

The Jacada Visual IVR software is implemented as applications for Apache Tomcat application server. For the compliance test environment the Jacada Visual IVR Evaluation Kit is deployed which includes an Apache Tomcat server, Oracle Sun Java run-time environment, and MongoDB. The Evaluation Kit is installed on the same server that Avaya Aura® Orchestration Designer resides, with one instance of Apache Tomcat server supporting Jacada Visual IVR applications and a second Apache Tomcat server supporting Avaya Aura® Orchestration Designer.

Jacada Visual IVR supports VoiceXML applications. In order to run VoiceXML applications in Jacada Visual IVR environment, the applications have to be enhanced with Jacada annotations, which are instructions used to parse VoiceXML language into Jacada Visual IVR elements.

To verify Jacada Visual IVR's interoperability with Avaya Aura® Orchestration Designer, four sample applications that originally came with Avaya Aura® Orchestration Designer (CollectTicketInfo, DynamicPhraseLoading, StockAndWeatherAudio, and MyCreditCard) have been enhanced with Jacada annotations and used in the test. The main integration point between Jacada Visual IVR and Avaya Aura® Orchestration Designer is the VoiceXML language. The VoiceXML language supports various elements such as prompts, grammars, forms, JavaScripts, variables, etc. The four sample applications only cover a subset of the usages of those elements. Any aspects of the VoiceXML language usage that are not covered by the four sample applications are outside the scope of the compliance test. In addition to the four sample applications that demonstrated interoperability between Jacada Visual IVR and Avaya Aura® Orchestration Designer, Jacada also provided a voice based application (VIVRTransferExample) to demonstrate limited interoperability with Avaya Aura® Experience Portal.

The target customers of Jacada Visual IVR are the enterprises who have deployed Avaya Aura® Experience Portal. Under agreement with Avaya product management, this solution would only be supported for customers who use Avaya Aura® Experience Portal in conjunction with the Jacada Visual IVR.

2. General Test Approach and Test Results

This section describes the compliance testing approach used to verify Jacada Visual IVR integration with Orchestration Designer and Experience Portal as well as the test results.

The compliance test used four sample Orchestration Designer applications originally delivered with Orchestration Designer to perform the test. The applications were enhanced with Jacada annotations to allow them to generate VoiceXML pages that are suitable for Jacada Visual IVR's processing. While three of the four sample applications were enhanced by Jacada, one sample application was enhanced by the Avaya test engineer.

The main focus of the test was to make sure that by using Jacada Visual IVR to access an Orchestration Designer application, the user could get the same experience as what they would get through Experience Portal. The experience included prompts, input options, responses, feature operations, and sequence of actions. The test involved identifying all the paths of the application menu and using web based interactions to exercise all the paths. Conditions where the user responses were invalid were also verified.

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.

2.1. Interoperability Compliance Testing

The interoperability compliance testing included feature and serviceability testing. The feature test cases were performed manually based upon the following steps:

- From a web browser on a PC or a mobile device enter the URL that is associated with a Jacada sample application.
- Interact with the sample application to exercise all the possible paths of the application tree.
- Use a PSTN phone to access the same application with all the same paths and verify that the user experience are the same between the two approaches.
- Along the application tree, enter invalid values and verify that the responses from the two approaches are the same.
- Since Orchestration Designer supports four different grammar types, test cases were repeated with each grammar setting.
- Repeat the above steps for each sample application.

The serviceability testing focused on verifying the ability of the Jacada/Orchestration Designer server to recover from adverse conditions, such as network outages and server reboots.

2.2. Test Results

All test cases passed during the compliance test. However, some behavior differences were observed between Experience Portal and Jacada Visual IVR when they were used to access the same sample application in Orchestration Designer. The differences were acceptable differences as they reflected how people operated in the voice world versus web world. The following were the observed differences.

- When a user is traversing the StockAndWeatherAudio sample application via Experience Portal, the user can return to the main menu at any point by entering a "*" key or saying "main menu". Jacada Visual IVR does not support the "*" key or "main menu" command. But the user can achieve the same purpose by killing the current interaction and start a new interaction from the web browser.
- When a user is in the middle of receiving stock or weather information from the StockAndWeatherAudio sample application via Experience Portal, the user can immediately move to the next prompt using a "Stop" or "Next" command. That is a part of the barge-in feature that Orchestration Designer supports. It is an implicit and optional step because if the user does not provide the command, the next prompt will be played after the stock or weather information is fully played. With Jacada Visual IVR the step is explicit and mandatory. The user has to choose "Stop" or "Next" before the next prompt is displayed.
- Jacada Visual IVR does not accept "two thousand thirteen" or "two thousand fourteen" as a way to specify a year in grammar as Experience Portal does. Jacada requires the customer to modify the grammar to use numbers such as "2013 and "2014" as they are more user-friendly.

During the compliance test the following issues were uncovered and subsequently fixed in Visual IVR Release 1.0 Build 11 by Jacada.

- SRGS-SISR, Nuance, and IBM grammar types were not supported.
- Pop-up window did not show up when the call button was pressed.
- Application processing errors for CollectTicketInfo (adult vs child in display, allowed "1" when "one" was expected) and MyCreditCard applications (credit card number validation error).

2.3. Support

For technical support on Jacada Visual IVR, contact Jacada via phone, email, or internet.

• **Phone:**	(888) 261-7618
• **Email:**	support@jacada.com
• **Web:**	http://support.jacada.com

3. Reference Configuration

Figure 1 illustrates the configuration used for testing. The Avaya products used included Experience Portal, Orchestration Designer with Apache Tomcat Server, Communication Manager, Session Manager, System Manager, and a number of phones. The Orchestration Designer and Apache Tomcat software were installed on a VMWare based virtual machine residing on a server blade. Jacada Visual IVR and a second instance of Apache Tomcat server were also installed on the same virtual machine. Experience Portal interfaced with Communication Manager via both H.323 and SIP connections. Session Manager and System Manager were present to provide the SIP connections.

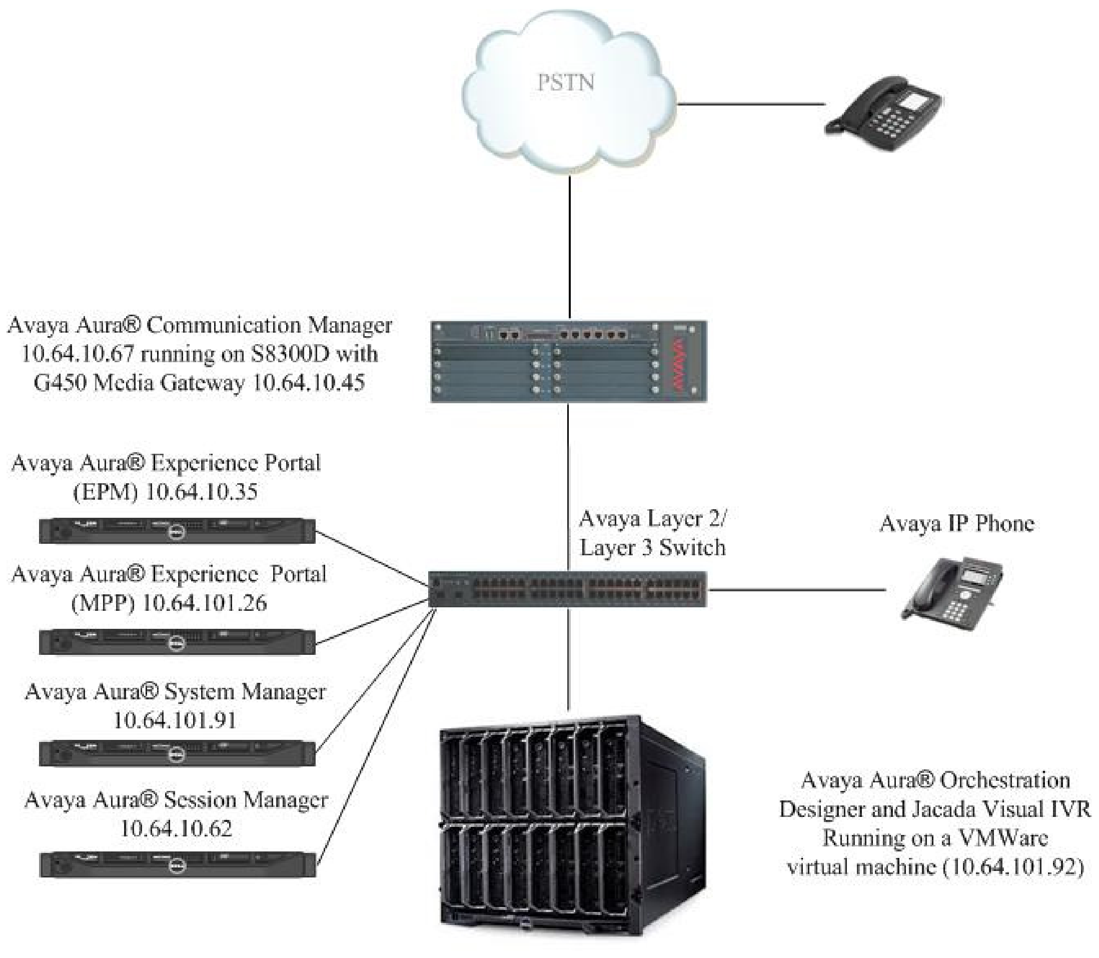

PSTN
Avaya Aura® Communication Manager 10.64.10.67 running on S8300D with G450 Media Gateway 10.64.10.45
Avaya Aura® Experience Portal (EPM) 10.64.10.35
Avaya Layer 2/ Layer 3 Switch
Avaya IP Phone
Avaya Aura® Experience Portal (MPP) 10.64.101.26
Avaya Aura® System Manager 10.64.101.91
Avaya Aura® Session Manager 10.64.10.62
Avaya Aura® Orchestration Designer and Jacada Visual IVR Running on a VMWare virtual machine (10.64.101.92)

Figure 1: Configuration with Avaya Aura® Experience Portal and Jacada Visual IVR

4. Equipment and Software Validated

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

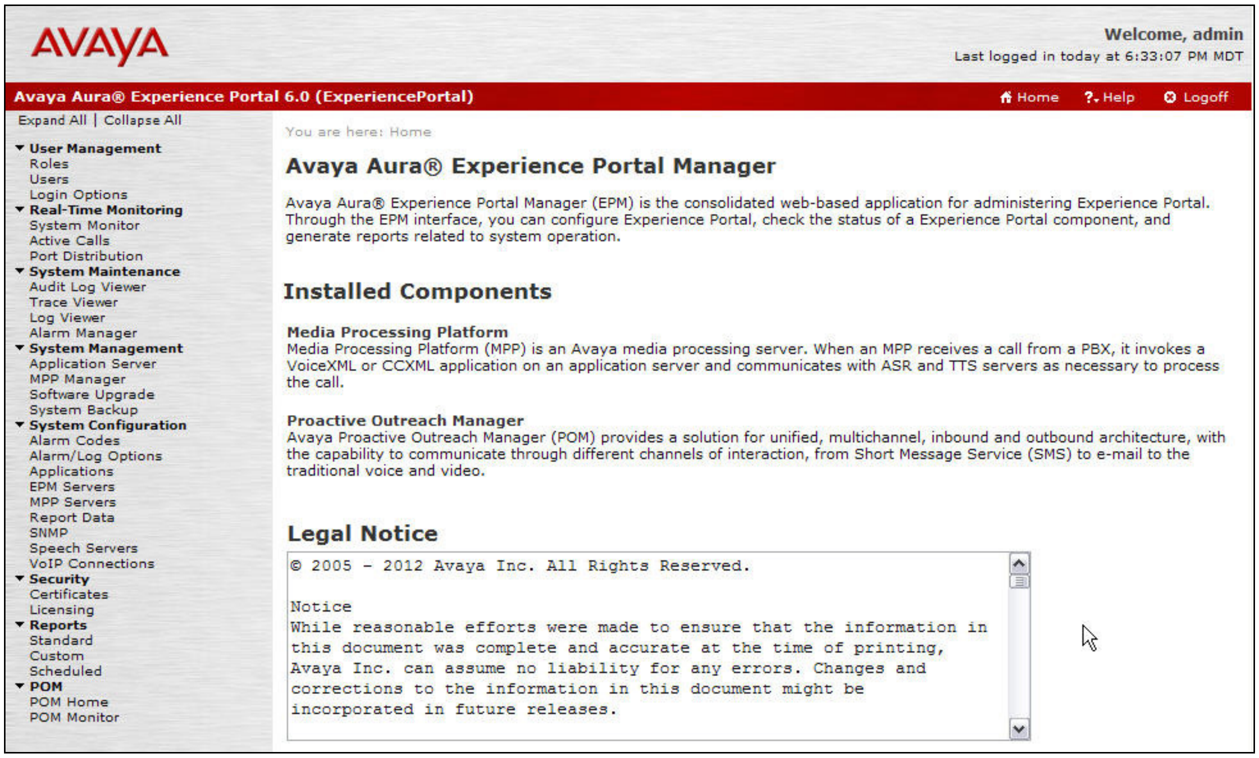
Equipment/Software	Version
Avaya Aura® Orchestration Designer with Apache Tomcat 6.0.29 running on a VMWare Virtual Machine	R6.0
Avaya Aura® Experience Portal	R6.0 SP2
Avaya Aura® Communication Manager running on Avaya S8300D Server	Release 6.3 with patch 03.0.124.0-20553
Avaya G450 Media Gateway MGP MM710 T1 Module	 HW 1 FW 31.20.0 HW 04 FW 015
Avaya Aura® Session Manager running on HP DL360 G7	R6.3 SP2
Avaya Aura® System Manager running on a VMWare Virtual Machine	R6.3 SP2
Avaya 96x1 H.323 Telephones	Avaya one-X® Deskphone Release 6.2.3
Jacada Visual IVR with Apache Tomcat 7.0.37 running on the same Virtual Machine as Avaya Aura® Orchestration Designer	Release 1.0 Build 11

5. Configure Avaya Aura® Experience Portal

This section covers the administration of Experience Portal. It is assumed that the VoIP connections (SIP and H.323) between Experience Portal and Communication Manager have been established and necessary routing configurations on Communication Manager and Session Manager are in place. The following configuration steps are covered in this section:

- Configure Applications

Experience Portal is configured via the Experience Portal Manager (EPM) web interface. To access the web interface, enter **http://<ip-addr>** as the URL in an Internet browser, where **<ip-addr>** is the IP address of the EPM. Log in using appropriate credentials. The screen is shown as follows.



Avaya Aura® Experience Portal Manager

Avaya Aura® Experience Portal Manager (EPM) is the consolidated web-based application for administering Experience Portal. Through the EPM interface, you can configure Experience Portal, check the status of a Experience Portal component, and generate reports related to system operation.

Installed Components

Media Processing Platform
Media Processing Platform (MPP) is an Avaya media processing server. When an MPP receives a call from a PBX, it invokes a VoiceXML or CCXML application on an application server and communicates with ASR and TTS servers as necessary to process the call.

Proactive Outreach Manager
Avaya Proactive Outreach Manager (POM) provides a solution for unified, multichannel, inbound and outbound architecture, with the capability to communicate through different channels of interaction, from Short Message Service (SMS) to e-mail to the traditional voice and video.

Legal Notice

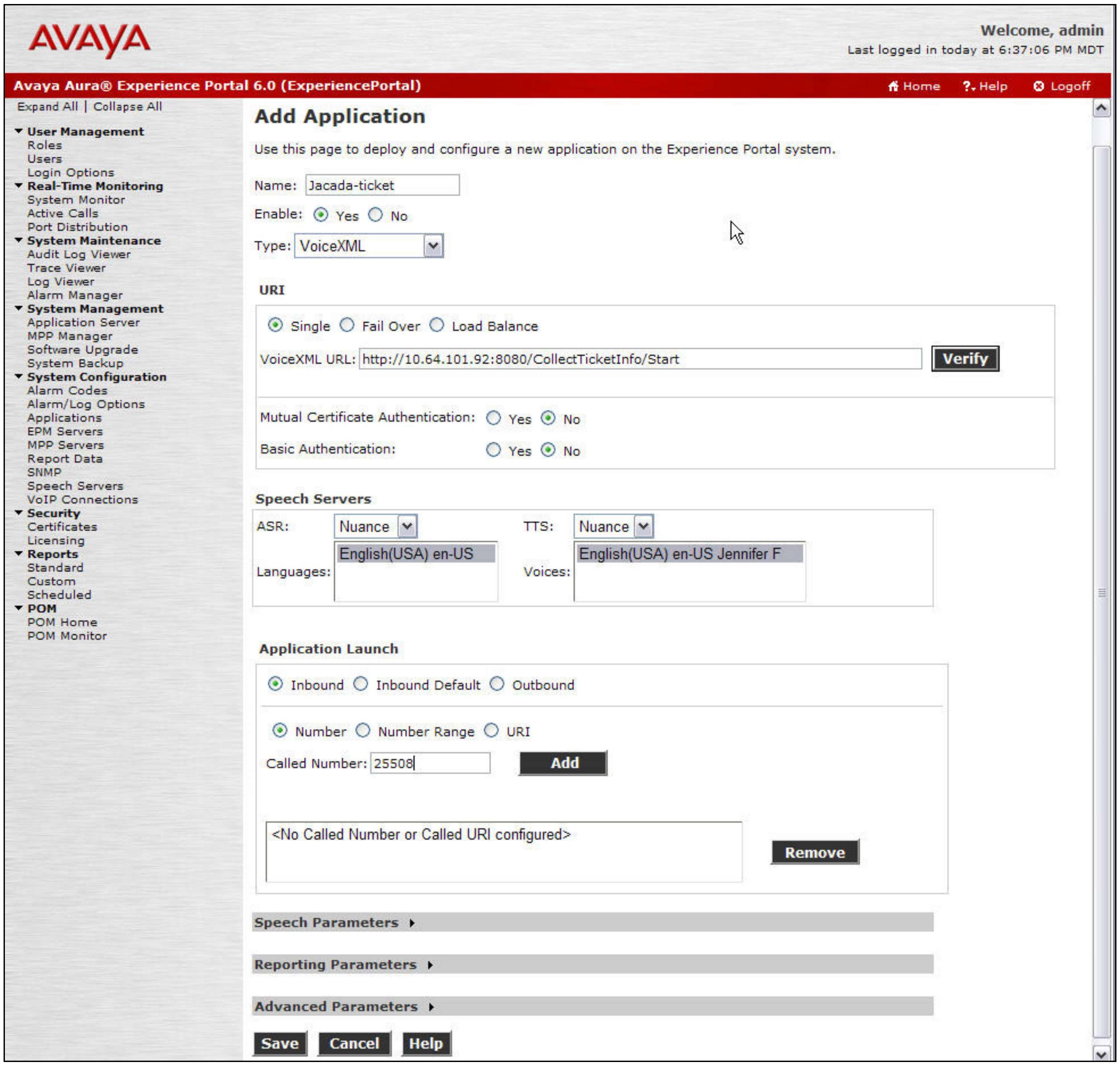
```
© 2005 - 2012 Avaya Inc. All Rights Reserved.

Notice
While reasonable efforts were made to ensure that the information in
this document was complete and accurate at the time of printing,
Avaya Inc. can assume no liability for any errors. Changes and
corrections to the information in this document might be
incorporated in future releases.
```

5.1. Configure Applications

From the left pane, click **System Configurations → Applications** to navigate to the **Applications** page and then click the **Add** button (not shown). The **Add Application** page is displayed. In the **Name** field, enter a descriptive name. Under the **URI** section, enter the URL to a Jacada sample application on the Tomcat Application Server in the **VoiceXML URL** field. In the **Application Launch** section, add a station extension and click **Add**. Use default values for the remaining fields. The screen below shows that the CollectTicketInfo application has been added with extension **25508**. Click **Save**.

Repeat the procedure for all the sample applications.



Add Application

Use this page to deploy and configure a new application on the Experience Portal system.

Name: Jacada-ticket

Enable: [✓ Yes] No

Type: VoiceXML

URI

[✓ Single] Fail Over Load Balance

VoiceXML URL: http://10.64.101.92:8080/CollectTicketInfo/Start

Mutual Certificate Authentication: Yes [✓ No]

Basic Authentication: Yes [✓ No]

Speech Servers

ASR: Nuance
Languages: English(USA) en-US
TTS: Nuance
Voices: English(USA) en-US Jennifer F

Application Launch

[✓ Inbound] Inbound Default Outbound

[✓ Number] Number Range URI

Called Number: 25508

<No Called Number or Called URI configured>

6. Configure Avaya Aura® Orchestration Designer

This section describes the configuration required on the Orchestration Designer server for supporting Jacada enhancements to the original Orchestration Designer sample applications. It is assumed that the four sample applications used in the compliance test have been imported into the Orchestration Designer environment already. In this section the following configuration steps on the Orchestration Designer server are covered:

- Install Jacada Elements
- Modify Prompts for Applications That Use Text-to-speech Technology
- Configure Transfer-To Number for CollectTicketInfo Application

6.1. Install Jacada Elements

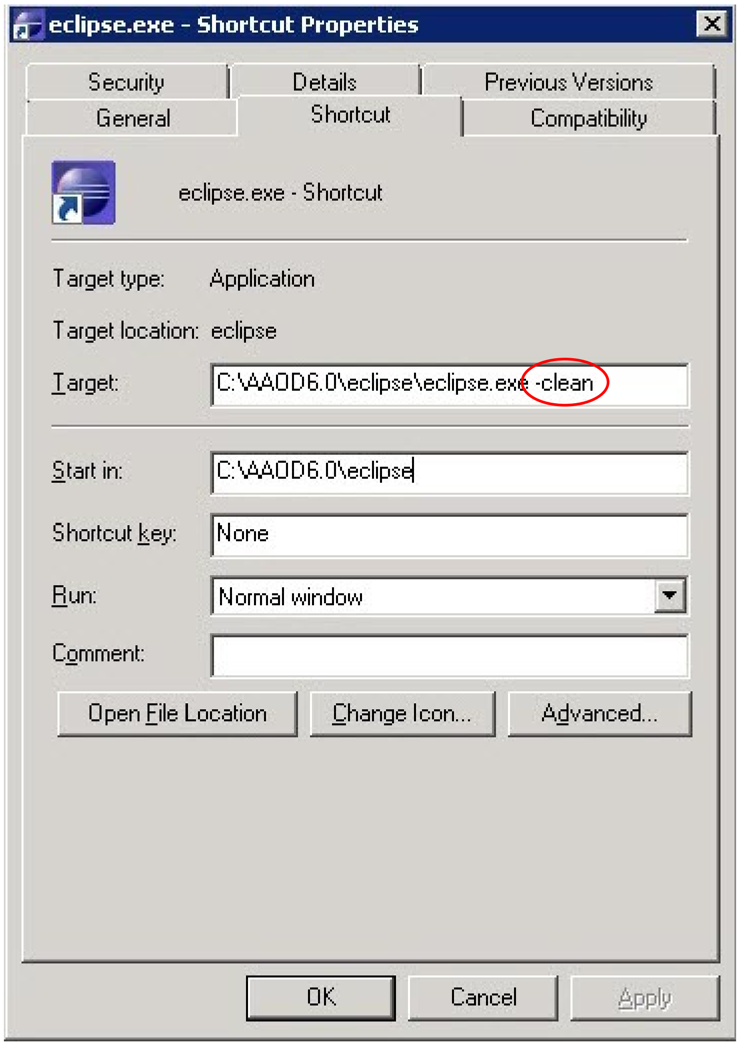
Two Jacada elements are required to be installed in the Orchestration Designer environment for supporting Jacada Visual IVR.

If Orchestration Designer is running, stop it by stopping the eclipse application.

Copy visualIVRAvaya.jar to **<TOMCAT_HOME>\lib** where **<TOMCAT_HOME>** is the root folder of the Apache Tomcat server software for Orchestration Designer.

Copy an Orchestration Designer plugin **com.jacada.visualivr.plugin_1.0.0.jar** to **<OD_HOME>\dropins** where **<OD_HOME>** is the root folder of the Orchestration Designer software.

Navigate to **<OD_HOME>\eclipse** and create a shortcut for eclipse.exe. Right click the shortcut and click **properties**. In the **Target** field, add "–clean" to the existing command. Click **OK** and double click the shortcut to start Orchestration Designer. Once Orchestration Designer is started, click **Tomcat → Start Tomcat** at the top tool bar to start Tomcat (not shown).



eclipse.exe - Shortcut

Target type: Application

Target location: eclipse

Target: C:\AAOD6.0\eclipse\eclipse.exe -clean

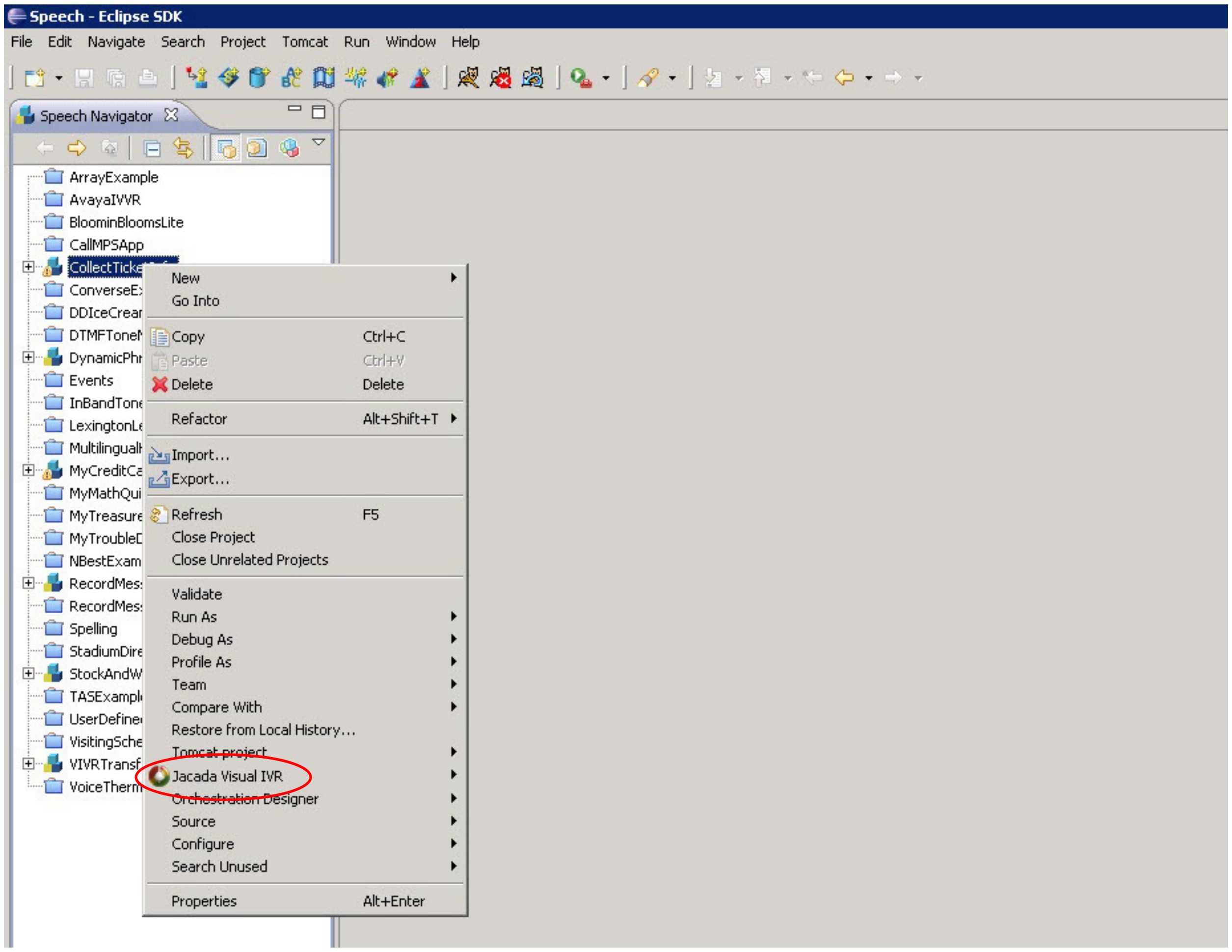
Start in: C:\AAOD6.0\eclipse

Shortcut key: None

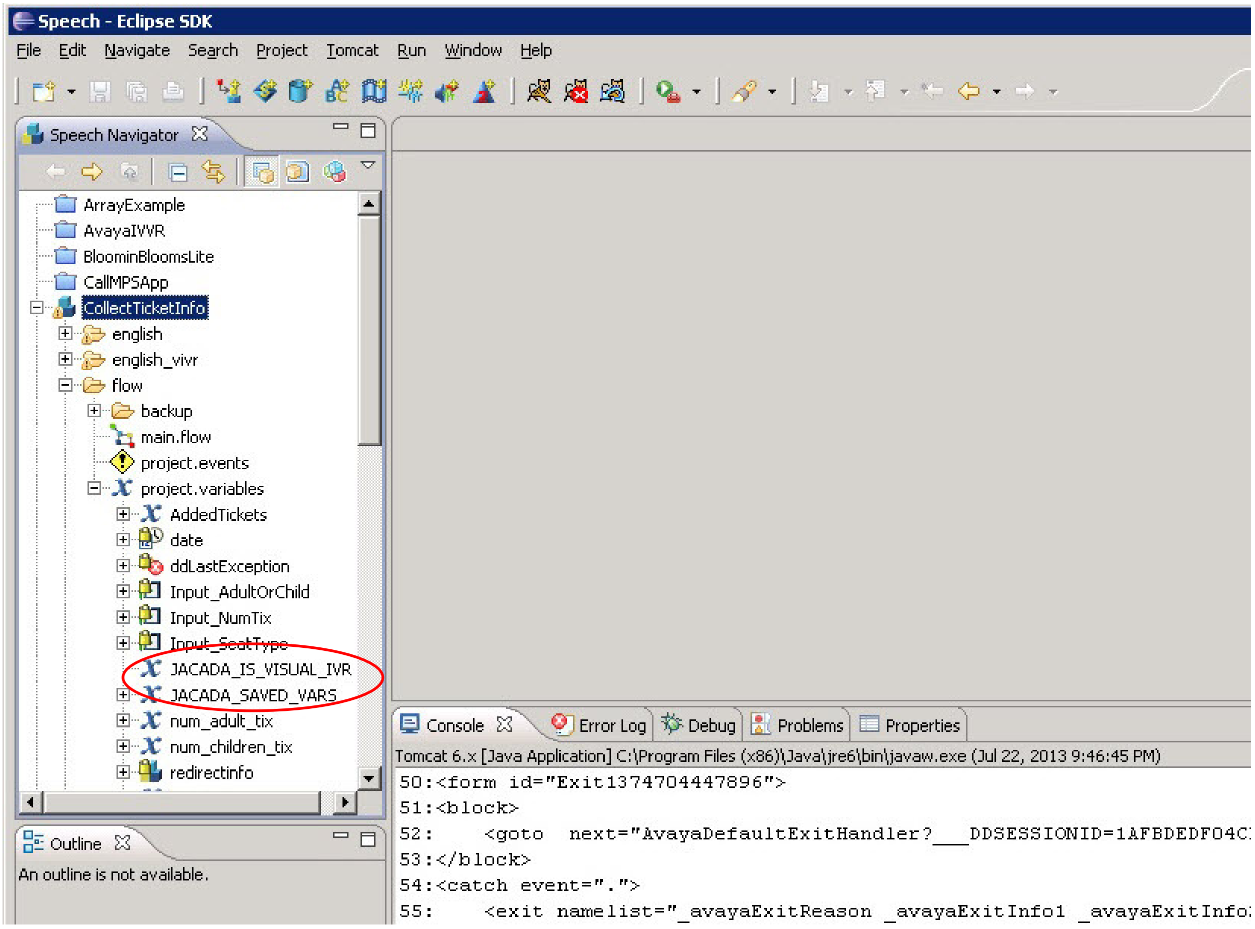
Run: Normal window

Comment:

Once the plugin is installed and Orchestration Designer is restarted, a new menu item **Jacada Visual IVR** will appear when right clicking each of the Orchestration Designer applications.



Two new Jacada related variables, **JACADA_IS_VISUAL_IVR** and **JACADA_SAVED_VARS**, will also appear for each of the Orchestration Designer applications in the **Speech Navigator** tab under the **flow→project.variables→Input_SeatType** subfolder.

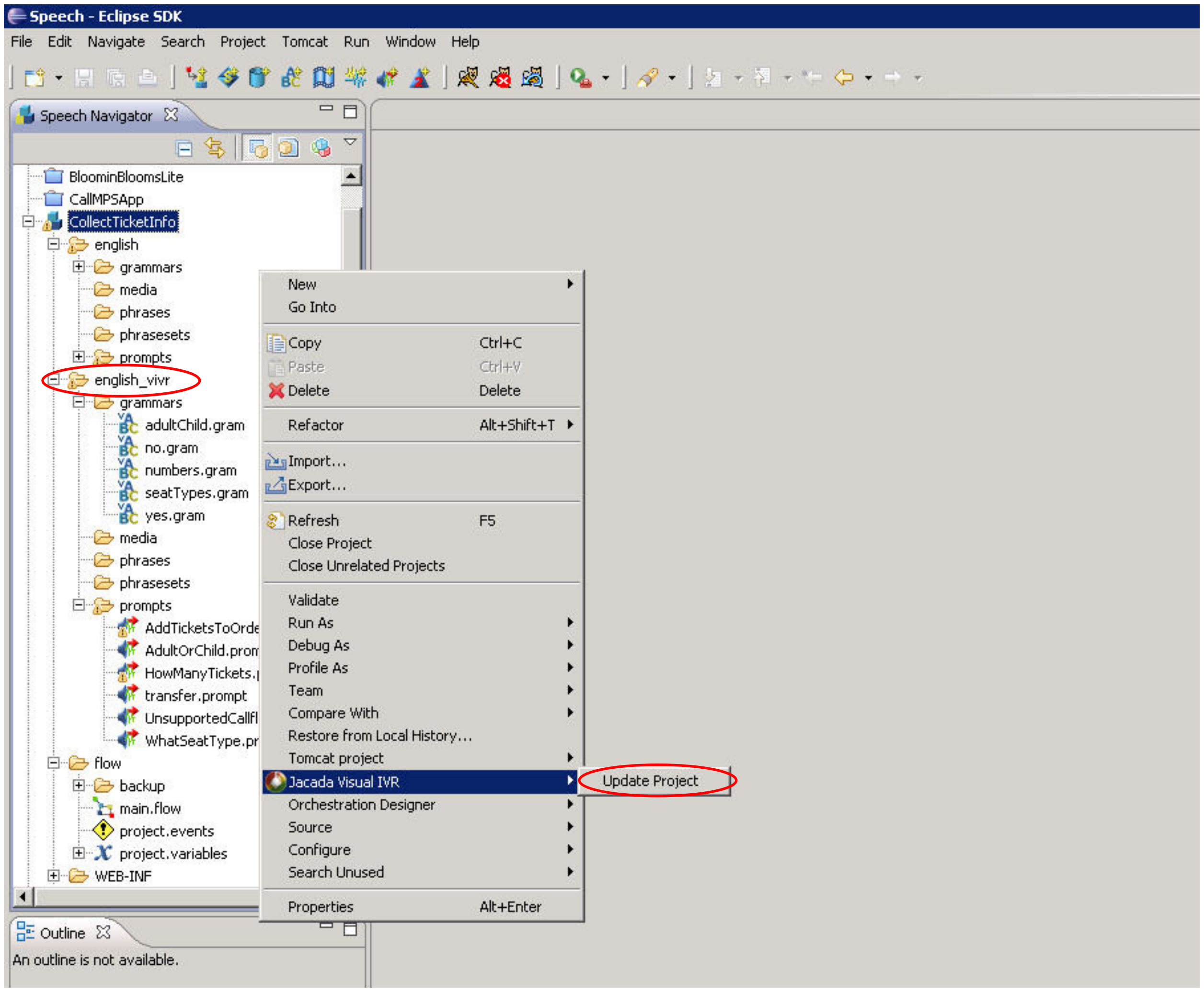


```
Tomcat 6.x [Java Application] C:\Program Files (x86)\Java\jre6\bin\javaw.exe (Jul 22, 2013 9:46:45 PM)
50:<form id="Exit1374704447896">
51:<block>
52:    <goto  next="AvayaDefaultExitHandler?___DDSESSIONID=1AFBDEDF04C
53:</block>
54:<catch event=".">
55:    <exit namelist="_avayaExitReason _avayaExitInfo1 _avayaExitInfo
```

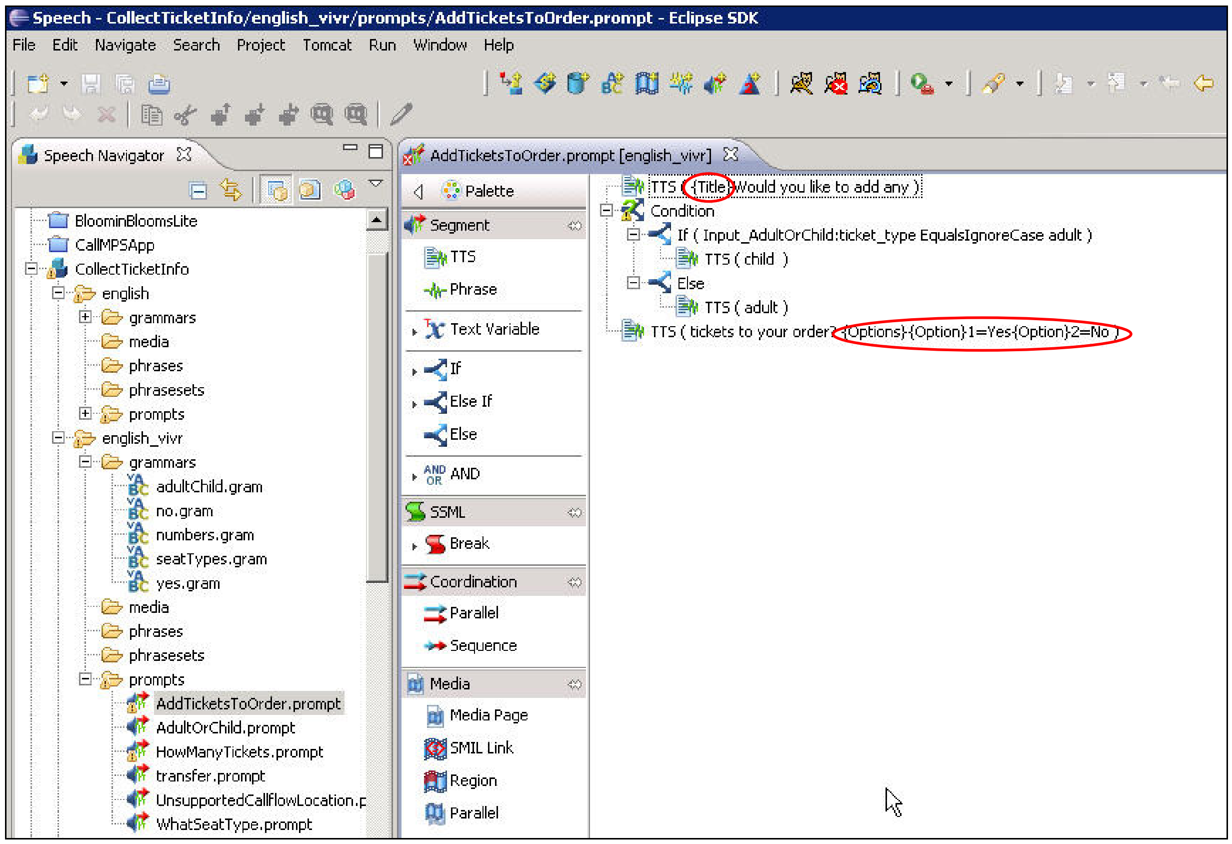
6.2. Modify Prompts for Applications That Use Text-to-speech Technology

In order for Orchestration Designer applications to work in Jacada environment, the applications have to be augmented to support Jacada annotations. For applications that use text-to-speech technology to play prompts, the following changes have to be made in Orchestration Designer:

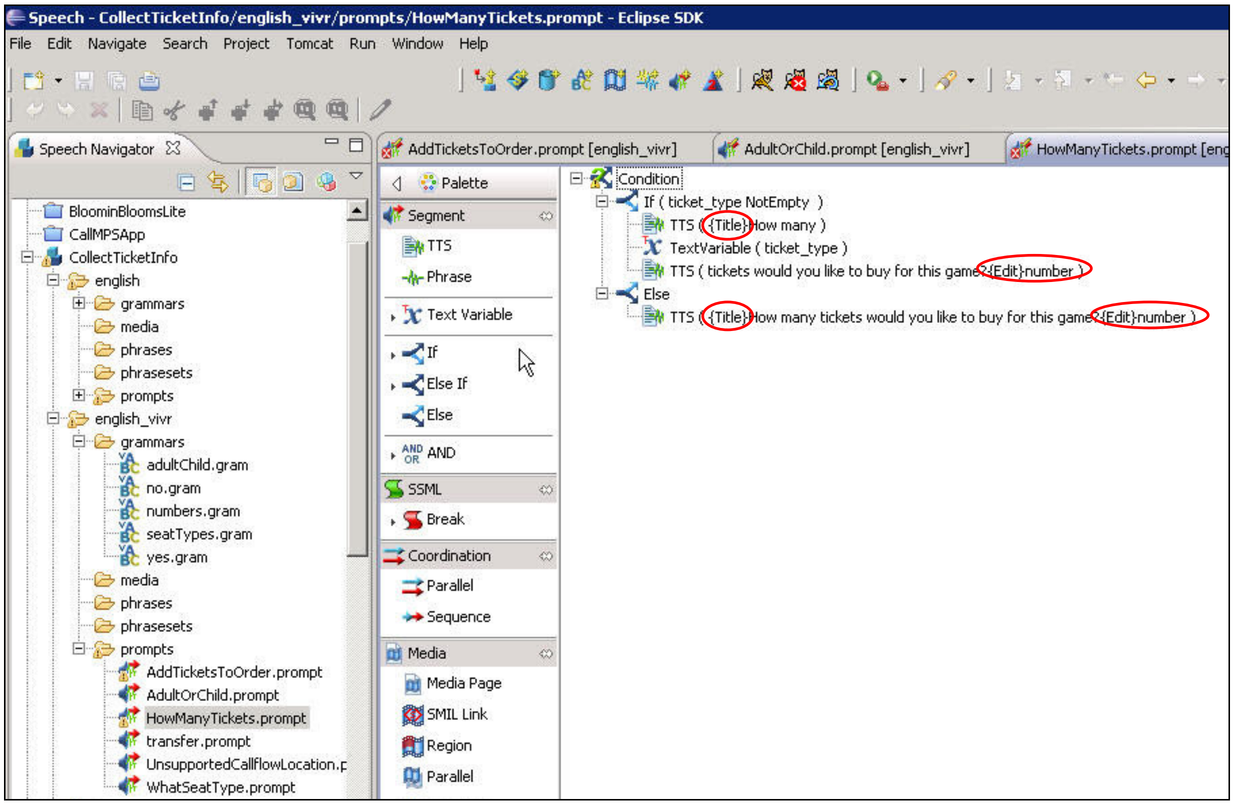
Under **Speech Navigator** tab, select an application and right click the mouse to generate a menu page. Navigate to **Jacada Visual IVR** and click **Update Project.** An **english_vivr** subtree will be created with the content replicating that of the **english** subtree. The screenshot below shows that an **english_vivr** subtree has been created for the CollectTicketInfo application.



For the prompts under **CollectTicketInfo → english_vivr → prompts**, add annotations according to Jacada guidelines and restart the Apache Tomcat server. The following two screenshots show a couple of prompts that have Jacada **Title**, **Edit**, and **Options** annotations added.



```
TTS ( {Title}Would you like to add any )
Condition
    If ( Input_AdultOrChild:ticket_type EqualsIgnoreCase adult )
        TTS ( child  )
    Else
        TTS ( adult )
TTS ( tickets to your order? {Options}{Option}1=Yes{Option}2=No )
```



```
Condition
    If ( ticket_type NotEmpty  )
        TTS ( {Title}How many )
        TextVariable ( ticket_type )
        TTS ( tickets would you like to buy for this game?{Edit}number )
    Else
        TTS ( {Title}How many tickets would you like to buy for this game?{Edit}number )
```

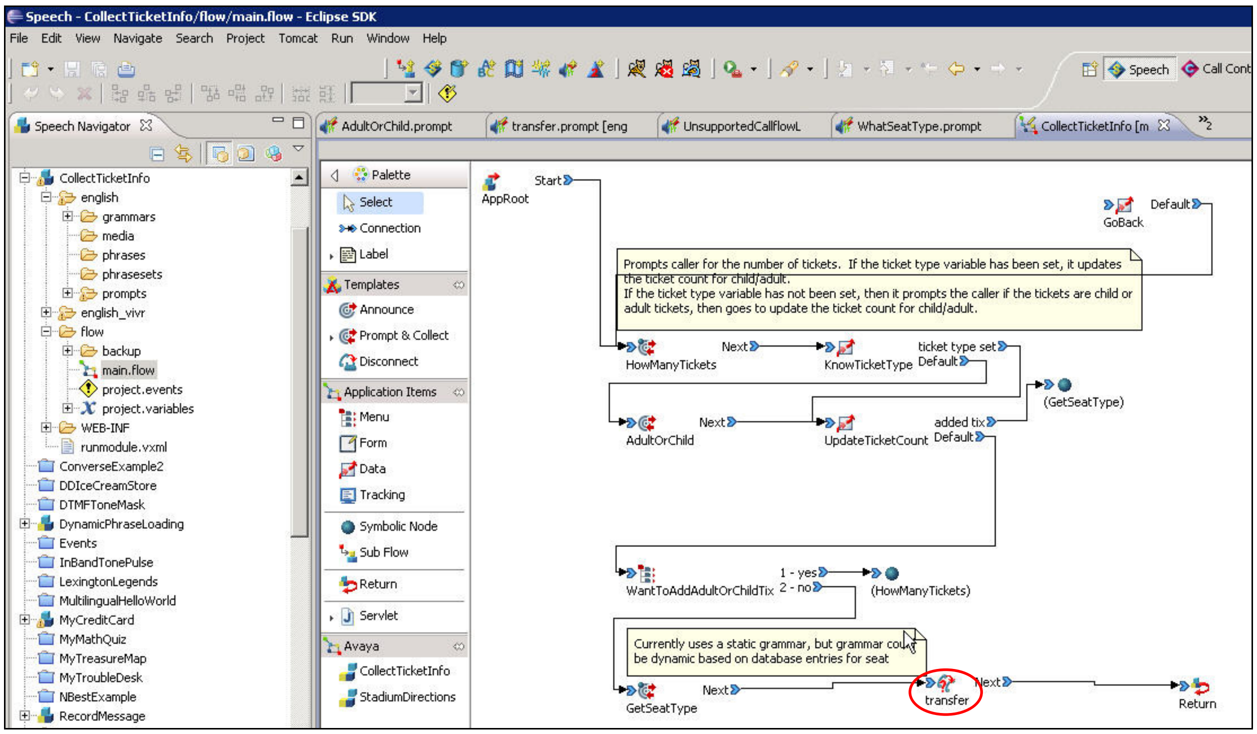
6.3. Configure Transfer-To Number for CollectTicketInfo Application

The CollectTicketInfo sample application was extended by Jacada to support a transfer-to-agent function. The transfer function works with a voice based application VIVRTransferExample, which was also provided by Jacada, to demonstrate interoperability between Jacada Visual IVR and Experience Portal.

The way the transfer-to-agent function and VIVRTransferExample application work is that once the customer has ordered tickets, he/she can opt to talk to an agent by clicking a **call** button. A pop-up window is displayed which shows the phone number for reaching the VIVRTransferExample application in the Experience Portal environment, along with a digit string that identifies the CollectTicketInfo interaction the customer just had with Jacada Visual IVR. The customer calls the VIVRTransferExample application and enters the digit string. The VIVRTransferExample application queries Jacada Visual IVR for the customer interaction data using the digit string as the key. The returned data includes a Transfer-To phone number for an agent group and an AAI (application to application information). The VIVRTransferExample application then issues a transfer command for the Experience Portal to transfer the call to the agent group.

Both the application phone number and Transfer-To phone number are configurable. The application phone number is configured in **Section 7.1**. The configuration of the Transfer-To number is specified below.

Under the **Speech Navigator** tab, double click **CollectTicketInfo → flow → main.flow** to display the main flow of the application in the upper right pane. Double click the **transfer** node.

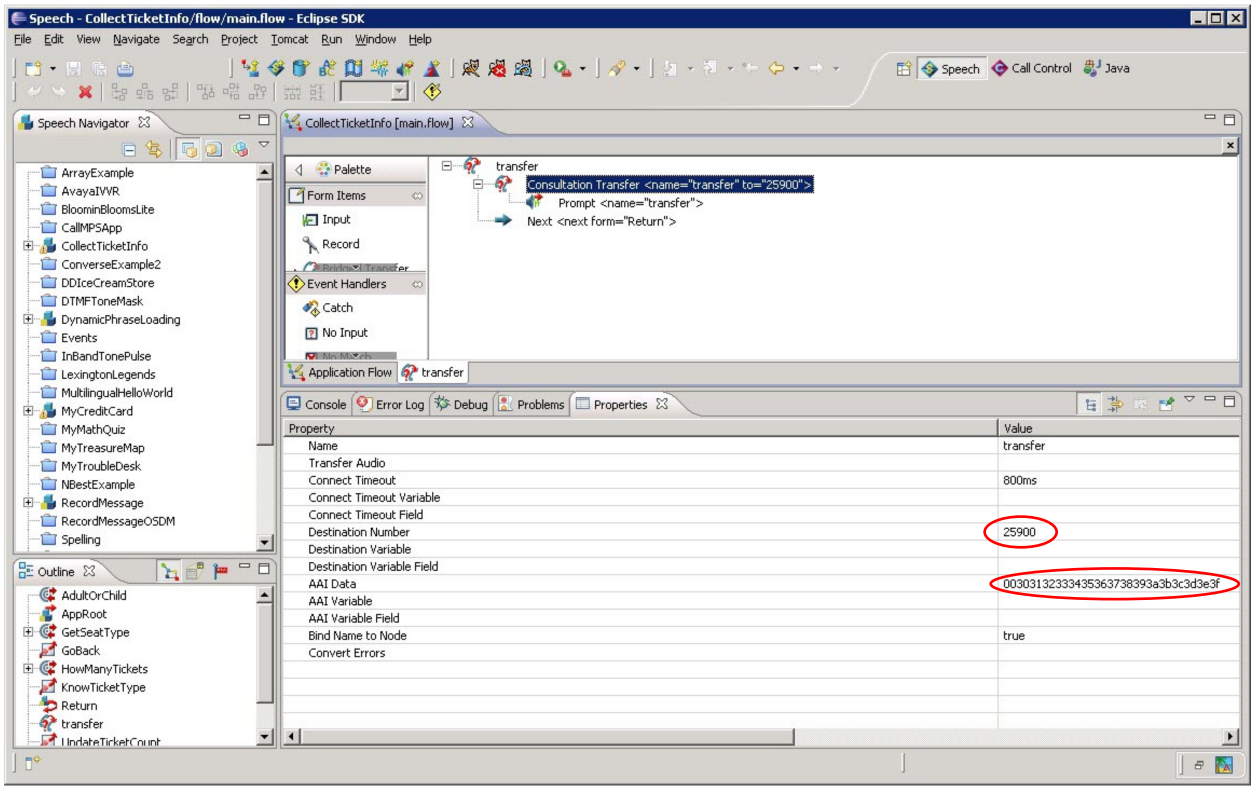


Prompts caller for the number of tickets. If the ticket type variable has been set, it updates the ticket count for child/adult.
If the ticket type variable has not been set, then it prompts the caller if the tickets are child or adult tickets, then goes to update the ticket count for child/adult.

Currently uses a static grammar, but grammar cou
be dynamic based on database entries for seat

The flow of the **transfer** node is displayed. Click **Window → Show View → Properties** in the top tools bar. The **Properties** window is displayed in the lower right pane. Set the **Destination Number** and **AAI Data** to desired values. Click **File → Save** to save the change.

Please note that the **AAI Data** is specified in hexadecimal form and prefixed with hex(00) in the reference configuration. In addition, H.323 connections between Experience Portal and Communication Manager do not support AAI data passing unless an AES connector function exists in Orchestration Designer (Note: AES connector is a connector to Avaya Aura® Application Enablement Services). Because the reference configuration does not include an AES connector, AAI data passing was only tested with SIP connections between Experience Portal and Communication Manager.


Speech - CollectTicketInfo/flow/main.flow - Eclipse SDK

```
transfer
  Consultation Transfer <name="transfer" to="25900">
    Prompt <name="transfer">
  Next <next form="Return">
```

Property	Value
Name	transfer
Transfer Audio	
Connect Timeout	800ms
Connect Timeout Variable	
Connect Timeout Field	
Destination Number	25900
Destination Variable	
Destination Variable Field	
AAI Data	00303132333435363738393a3b3c3d3e3f
AAI Variable	
AAI Variable Field	
Bind Name to Node	true
Convert Errors	

7. Configure Jacada Visual IVR

This section describers the configuration step for Jacada Visual IVR. The following procedures are covered:

- Configure Application.xml File
- Create Application-prompts.xml file
- Start Jacada Visual IVR

7.1. Configure Application.xml File

Each Orchestration Designer application used for Jacada environment requires an xml based configuration file to be created in the **<VIVR_HOME>\apache-tomcat-7.0.37\webapp\visualivr\vxml** folder where **<VIVR_HOME>** is the root folder of the Jacada Visual IVR software. The name of the xml file should be the same as the name of the Orchestration Designer application with .xml as the extension. The items to be configured include:

Name:	a descriptive name of the application
Start:	entry point of the Orchestration Designer application
AudioToTextTable:	name of the AudioToText translation table file. See **Section 7.2** for details
Manufacture:	**Avaya**
DefaultGrammarFormat:	type of speech recognition grammar
isCallEnabled:	flag to turn on the call function
PageTitle:	title of the interaction page
numberToDial:	number to dial for accessing the VIVRTransferExample application

The following is the configuration file for the CollectTicketInfo application.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Configuration Version="1">
<Name>CollectTicktInfo Example</Name>
<Start>http://localhost:8080/CollectTicketInfo/Start?JACADA_IS_VISUAL_IVR=Yes</Start>
<AudioToTextTable></AudioToTextTable>
<IVR>
<Manufacture>Avaya</Manufacture>
<Model></Model>
<Version></Version>
<DefaultGrammarFormat>application/x-nuance-osr</DefaultGrammarFormat><!-- Grammar
options use value field: 1. IBM(value: application/x-ibm) 2. Nuance
OSR(value:application/x-nuance-osr)3. SRGS-Literal(value: semantics/1.0-literals)4.
SRGS-SISR(value:semantics/1.0).-->
</IVR>
<Variables>
<Variable Name="isCallEnabled" Value="true" />
<Variable Name="isChatEnabled" Value="true" />
<Variable Name="PageTitle" Value="Tickets" />
<Variable Name="numberToDial" Value="17209772873"/>
</Variables>
</Configuration>
```

7.2. Create AudioToTextTable file

For applications that use audio recordings for prompts, an AudioToText translation table has to be created to translate the audio recordings into text with annotations. The file is also located in the **<VIVR_HOME>\apache-tomcat-7.0.37\webapp\visualivr\vxml** folder. The following is the AudioToText translation table developed for the StockAndWeatherAudio application. The **path** parameter specifies where the audio recordings (.wav files) are located. Each of the **prompt** lines following the **path** line specifies the mapping from a .wav file to the corresponding text.

```
<?xml version="1.0" encoding="utf-8"?>
<prompts>
       <directory
path="http://localhost:8080/StockAndWeatherAudio/data/english/phrases/">
              <prompt name="Stock_or_Weather_Menu.wav">{Title}Would you like to get a
stock quote or check the weather of a US
city?{Options}{Option}1=Stock{Option}2=Weather</prompt>
              <prompt name="Company_prompt.wav">{Title}Please speak the name of a
company{Dropdown}{Option}microsoft=Microsoft{Option}cisco=Cisco{Option}google=Google{O
ption}ebay=EBay{Option}united health group=United Health Group</prompt>
              <prompt name="Speak_Stock.wav">{Title}The common stock value of </prompt>
              <prompt name="is.wav"> is  </prompt>
              <prompt name="City_for_weather.wav">{Title}For which city would you like
to hear the weather?{Dropdown}{Option}san francisco=San Francisco{Option}san jose=San
Jose{Option}new york=Manhattan{Option}denver=Denver</prompt>
              <prompt
name="The_weather_in.wav">{Options}{Option}stop=Stop{Option}next=Next{Option}repeat=Re
peat{Title}The weather in </prompt>
              <prompt name="Another_Quote.wav">{Title}Would you like to get another
stock quote?{Options}{Option}1=Yes{Option}2=No{Option}3=Repeat</prompt>
              <prompt name="Another_City.wav">{Title}Would you like to hear the weather
in any other US cities?{Options}{Option}1=Yes{Option}2=No{Option}3=Repeat</prompt>
              <prompt name="Goodbye.wav">{Statement}Goodbye and thank you for
calling</prompt>
       </directory>
</prompts>
```

7.3. Start Jacada Visual IVR

A **startVIVR.bat** batch file is located in the root folder of the Jacada Visual IVR software. Once the configuration files and AudioToText translation table files are in place, start Jacada Visual IVR by running the **startVIVR.bat** file in a DOS window.

8. Verification Steps

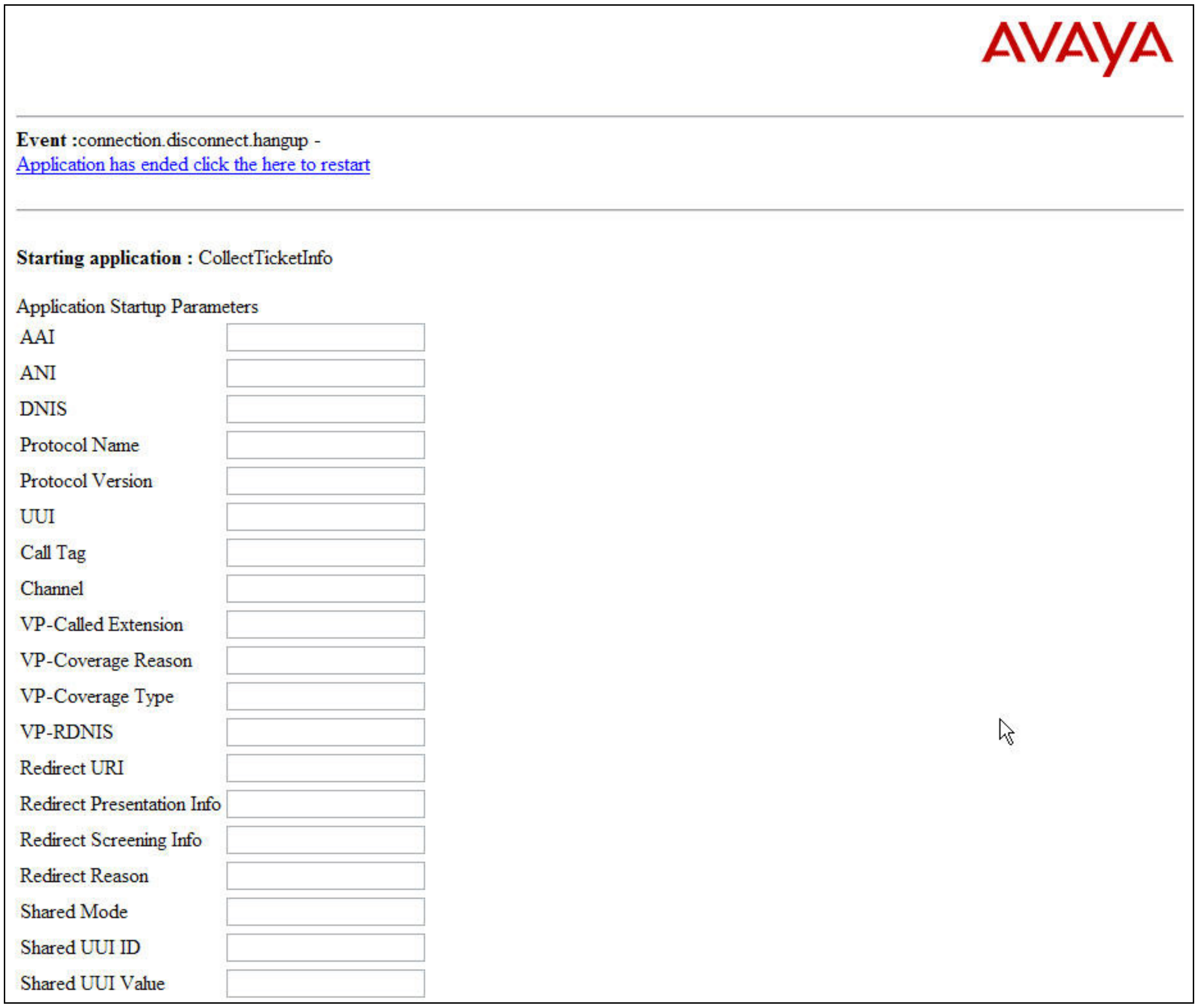
This section provides the steps to verify that a user can access the Jacada sample applications via Jacada Visual IVR. It also includes a few interaction pages from the CollectTicketInfo application as examples of what the user will see during interactions with Jacada Visual IVR.

8.1. Verify Start Line of Configuration File

From the configuration file of one of the sample applications, extract the content of the **start** line and put into the URL field of a web browser.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Configuration Version="1">
<Name>CollectTicktInfo Example</Name>
<Start>http://localhost:8080/CollectTicketInfo/Start?JACADA_IS_VISUAL_IVR=Yes</Start>
<AudioToTextTable></AudioToTextTable>
<IVR>
<Manufacture>Avaya</Manufacture>
<Model></Model>
<Version></Version>
<DefaultGrammarFormat>application/x-nuance-osr</DefaultGrammarFormat><!-- Grammar
options use value field: 1. IBM(value: application/x-ibm) 2. Nuance
OSR(value:application/x-nuance-osr)3. SRGS-Literal(value: semantics/1.0-literals)4.
SRGS-SISR(value:semantics/1.0).-->
</IVR>
<Variables>
<Variable Name="isCallEnabled" Value="true" />
<Variable Name="isChatEnabled" Value="true" />
<Variable Name="PageTitle" Value="Tickets" />
<Variable Name="numberToDial" Value="17209772873"/>
</Variables>
</Configuration>
```

Verify that that the following page is displayed as an indication that the application URL in the configuration file is correct.



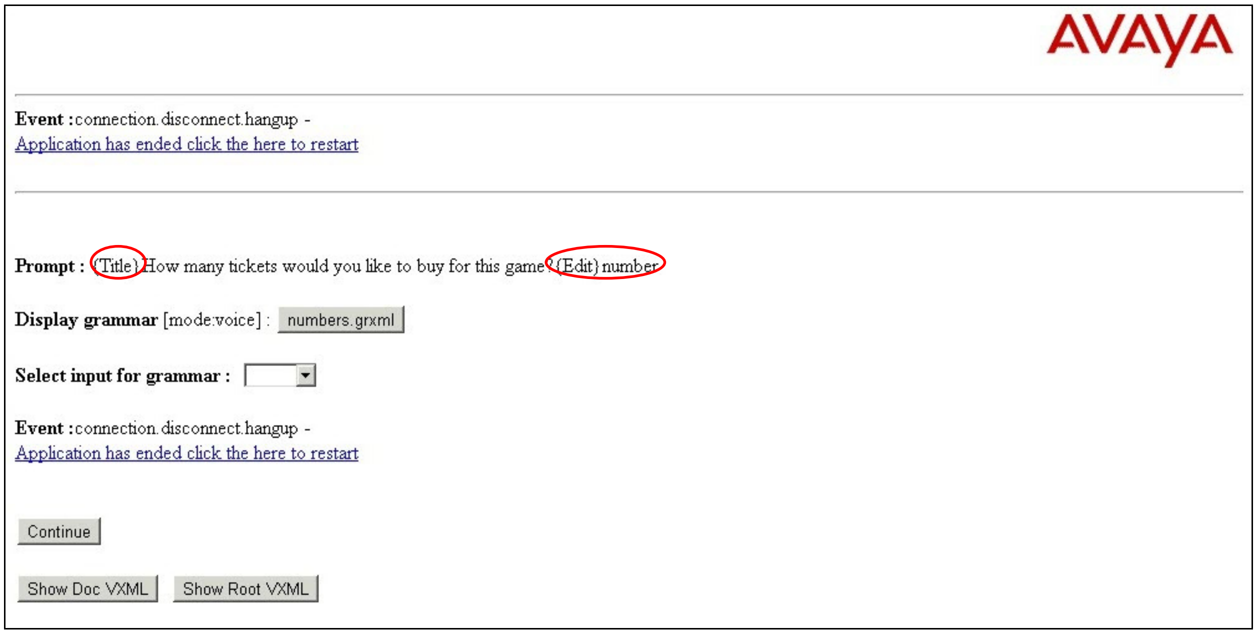
Event :connection.disconnect.hangup -
Application has ended click the here to restart

Starting application : CollectTicketInfo

Application Startup Parameters

AAI	
ANI	
DNIS	
Protocol Name	
Protocol Version	
UUI	
Call Tag	
Channel	
VP-Called Extension	
VP-Coverage Reason	
VP-Coverage Type	
VP-RDNIS	
Redirect URI	
Redirect Presentation Info	
Redirect Screening Info	
Redirect Reason	
Shared Mode	
Shared UUI ID	
Shared UUI Value	

For applications that were originally designed to use Text-to-Speech prompts, click the **Continue** button at the bottom of the page (not shown) to go through the application. Verify that Jacada annotations are displayed as a part of some prompts.


Event : connection.disconnect.hangup -
Application has ended click the here to restart

Prompt : {Title} How many tickets would you like to buy for this game? {Edit} number

Display grammar [mode:voice] :

Select input for grammar :

Event : connection.disconnect.hangup -
Application has ended click the here to restart

8.2. Verify Jacada Visual IVR Interactions

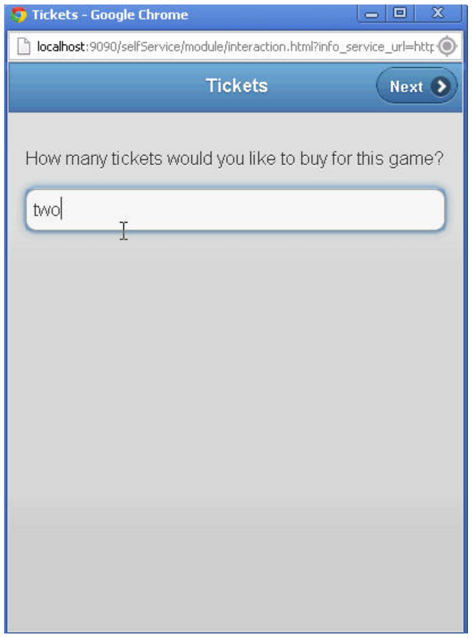
To access the Jacada sample applications via Jacada Visual IVR, the user has to use a URL of the following format:

http://localhost:9090/selfService/?interaction=51c691cb1a7ea0d1a5d79bb5&Project+Name=CollectTicketInfo&mode=vivr&windowMode=new&windowTitle=collect

The URL includes the port (9090) of the Apache Tomcat server that Jacada Visual IVR runs on and the name of the application (CollectTicketInfo). The name of the application has to match the name of the configuration file for the application.

Access to Jacada applications requires the use of a web browser that supports HTML5, for example, Google Chrome.

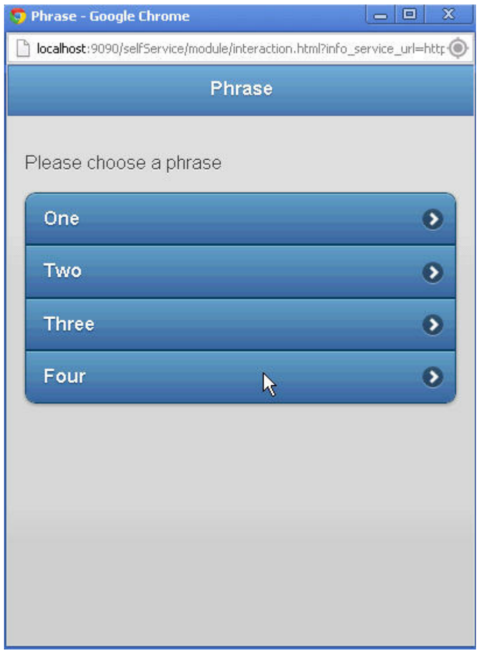
The URL will trigger a pop-up window with prompts. The user is expected to enter a response in a space following the prompt,


Tickets - Google Chrome
localhost:9090/selfService/module/interaction.html?info_service_url=http
Tickets
Next

How many tickets would you like to buy for this game?

two

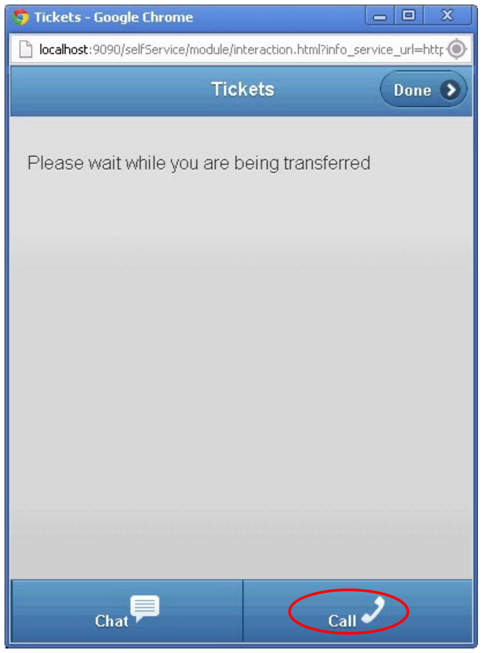
or select an option from a list like the following:


Phrase - Google Chrome
localhost:9090/selfService/module/interaction.html?info_service_url=http
Phrase
One
Two
Three
Four

Please choose a phrase

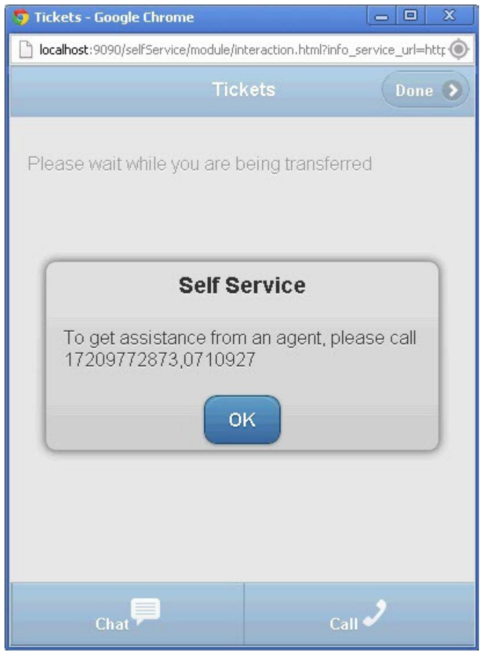
8.3. Verify Function of the Call Button

At the end of the interactions with the CollectTicketInfo application, the following page is displayed:


Tickets - Google Chrome
localhost:9090/selfService/module/interaction.html?info_service_url=http
Tickets
Done
Chat
Call

Please wait while you are being transferred

If the user clicks the **call** button, the following page is displayed which includes the phone number for the VIVRTransferExample application and a digit string for accessing customer interaction data.


Tickets - Google Chrome
localhost:9090/selfService/module/interaction.html?info_service_url=http
Tickets
Done
OK
Chat
Call

Please wait while you are being transferred

Self Service

To get assistance from an agent, please call 17209772873,0710927

9. Conclusion

These Application Notes describe the configuration steps required to integrate the Jacada Visual IVR application with Avaya Aura® Orchestration Designer and Avaya Aura® Experience Portal. All feature and serviceability test cases were completed successfully with observations documented in **Section 2.2**.

10. Additional References

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

- Avaya Aura® Orchestration Designer Getting Started with Orchestration Designer Release 6.0
- Avaya Aura® Orchestration Designer Installation Notes
- Avaya Aura® Orchestration Designer Sample Applications, Version 6.0
- Jacada Visual IVR Installation and Quick Start Guide, Version 1.0
- Jacada Visual IVR Administration Guide, Version 1.0