AVAYA

Avaya Aura® Orchestration Designer
Release 6.0.0.9.01
Release Letter

(Supersedes All Previous Releases)

Date: August 4, 2011

Table of Contents

1 Release Information

This is the GA candidate build of Orchestration Designer.

1.1 Avaya Aura® Contact Center Tool Information

The AACC tool build version in this release is sce30_1304.
.

1.2 Release Overview

The primary themes of the 6.0 release are:

- Developer Usability and Productivity
    - Support for compound conditionals in the Call Flow and Prompt Editors
    - Date and Time based routing in the Call Flow Editor
    - Numerous Enhancements (see MR list for descriptions)
- Simulation MRCP Support
    - Customers will be able to use the simulator in conjunction with Nuance and Loquendo’s TTS and ASR
- GRIP Requests
    - Improved efficiency of subdialog/modules
    - Specifying prompts and grammars by a variable
    - Applications shall support deploy/undeploy without requiring server restart
    - IPv6 support
- MPS Interoperability and Investment Protection
    - Add a palette item for calling MPS developer applications
    - New sample application to demonstrate calling an MPS Developer application
    - Add support for OD applications to be launched by a MPS Developer application
    - Add support for MMF audio files
    - Rename CTI connector to AES
- Integration with new Orchestration Designer Framework
    - New splash screens and rebranding of umbrella framework
    - Support co-residency with AACC SCE tool
    - New documentation to describe Orchestration Designer components, support, etc.
- Currency with 3rd party products
    - Oracle WebLogic 11g
    - Eclipse 3.6
    - Avaya Aura® Experience Portal 6.0

2 Features

This section describes the status of the features that are available in this release to SV.

2.1 New Features included in this delivery

N/A

2.2 Requirements not included in this delivery

N/A

2.3 Features/Requirements with Limitations/Restrictions in this delivery

- There is a conflict with the Axis2 component installed with Websphere 7 and the Voice Portal Configurable variables and reporting features causing them not to function properly.
- For the development environment, you can use the 64 bit version of the OS, but you must use the 32 bit version of the other pre-requisites, Java and Eclipse. For the runtime, you can use 64 bit Java.

- MRCP Limitations/Notes:
    - When using MRCP for ASR in the simulator, make sure you select the right grammar compatibility depending on the speech server you are using. Select your project and goto Project -> properties -> Orchestration Designer-> Speech and select the needed grammar compatibility.
    - Simulation with MRCP using speech servers is slower than simulation using Microsoft SAPI. This is because with MRCP we contact remote speech servers while Microsoft SAPI is local to the system. Speech bargein is slower due to the same reason.
    - When MRCP ASR is enabled, you cannot provide text input for speech recognition.
    - MRCP V2 is not supported in this release.
    - When using MRCP for ASR, spoken input should be provided to the simulator via a microphone only after you see the message "Waiting for ASR input" in the simulator.
    - It is assumed that your third party TTS and ASR speech servers are installed on the same machine and thus have the same IP address and port number.

- Eclipse version 3.5.2 or higher is required for this build.

2.3.1 Nuance NDM version 5.2+

Nuance has changed the format of how you invoke NDM's starting with version 5.2.0 and greater. The VXML that Dialog Designer generates to call the NDM will result in the browser returning an error indicating the NDM was not found (a 404 error). To remedy this, you can invoke the NDM manually by following these steps:

Make sure you have deployed ndm-core.war v.5.2.0 or greater onto your appserver.
Add a VXML servlet node to your flow
Edit the Java code in your VXML servlet node and locate the pregenerated method:

```
    public void markupLanguageGeneration(java.io.PrintStream out, com.avaya.sce.runtime.Submit submit, com.avaya.sce.runtimecommon.SCESession mySession)
```

Enter the following code

```
1. out.println("<var name=\"SDdigits___returncode\"/>");
2. out.println("<var name=\"SDdigits___returnvalue\"/>");
3. out.println("<var name=\"dmname\" expr=\"'mydigits'\"/>");
4. out.println("<var name=\"osdm\" expr=\"'digits'\"/>");
5. out.println("<subdialog name=\"dmSessionStart\" src=\"/ndm-core/controller/sessionstart\" method=\"get\"/>");
6. out.println("<subdialog name=\"SDdigits\" src=\"/ndm-core/controller/ndm\"  method=\"get\" namelist=\"dmname osdm\">");
7. out.println("<filled>");
8. out.println("<assign name=\"SDdigits___returncode\" expr=\"SDdigits.returncode\"/>");
9. out.println("<assign name=\"SDdigits___returnvalue\" expr=\"SDdigits.returnvalue\"/>");
10. out.println("</filled>");
11. out.println("</subdialog>");
12. out.println("<subdialog name=\"dmSessionEnd\"  src=\"/ndm-core/controller/sessionend\"
method=\"get\"/>");
13. submit.addToSubmitList("SDdigits___returncode");
14. submit.addToSubmitList("SDdigits___returnvalue");
15. super.markupLanguageGeneration(out, submit, mySession);
```

Lines 1 and 2 define the return values you are expecting
Lines 3 and 4 define variables passed into the NDM call. In this case dmname is an arbitrary unique name and osdm indicates the name of the osdm to invoke.
Line 5 calls SessionStart in the ndm-core. This is a mandatory call that must be made once before calling any NDM's or they will not work.
Line 6 is our call into the "digits" NDM.

Lines 7-10 retrieve the return value from the call and assign them to our variables. Each NDM has their own return values.
Line 12 Ends the session. You must call this when you are done invoking your NDM's

NOTE: Please consult Naunce's NDM handbook for the version you are using to get more complete details on using NDM's.

2.3.2 Orchestration Designer Debugger support (Properties Editor style –PDC only)

By default, breakpoints are enabled for all Data Node call flow items. If you do not want Orchestration Designer developers to be able to place breakpoints on your contributed items, you must implement supportsBreakPoints() and return **false**. The default implementation of supportsBreakpoints() returns **true**.

Pluggable connectors that wish allow breakpoints on their contributed items need to change their code generation. The change is to set the debug ID on the action. The debug ID is the call flow object ID. It is a simple change to the code generation when you generate the closing parentheses.

Change this:

```
sb.append(“));”);
```

to this:

```
sb.append(“).setDebugId(“+item.getFlowObjectId()+”));”);
```

This results in the code generation, for example, changing from this:

```
//{{START:CODEGEN:EXTENSIONPOINT:com.avaya.sce.pdc.pe.advancedmath.pi
actions.add(new com.avaya.sce.pdc.pe.advancedmath.Pi(“session”, “exitInfo1”));
//}}END:CODEGEN:EXTENSIONPOINT:com.avaya.sce.pdc.pe.advancedmath.pi
```

to this new code generation:

```
//{{START:CODEGEN:EXTENSIONPOINT:com.avaya.sce.pdc.pe.advancedmath.pi
actions.add(new com.avaya.sce.pdc.pe.advancedmath.Pi(“session”, “exitInfo1”).setDebugId(9));
//}}END:CODEGEN:EXTENSIONPOINT:com.avaya.sce.pdc.pe.advancedmath.pi
```

In addition, the runtime implementation of your pluggable connector must implement Idebugging. The simplest way to add this is to extend the Debugging class. For example,

Prior to adding debugger upport:

```
public class Pi implements IpluggableExecutable {
```

After adding debugger support:

```
public class Pi extends Debugging implements IpluggableExecutable {
```

The base class Debugging implements the Idebugging interface.

2.3.3 New Web Service Connector

Axis 2 jar files
The new web service connector is based on Axis 2 version 1.5. To limit the number of jar files in runtimesupport.zip we have chosen not to include the full set of jar files from the Axis 2 distribution in DD. Instead we have includes a set of jar files which cover “normal usage” of the web service connector. Should it happen that you get Class Not Found exception while executing a web service operation you can download

the full set of Axis 2 jar files from http://ws.apache.org/axis2/download/1_5/download.cgi and put them in your tomcat lib folder.

SOAP message signing
The new web service connection has an option for signing the soap message. Generation of this signature is based on a key stored in a certificate store. It is currently not supported reading that key out of a keystore generated by the DDAdmin keystore management. Instead it is recommended using the keytool from your java distributation like this:

keytool –genkey –alias privkey –keystore c:\privkeystore –dname “cn=privkey” –keypass password1 –storepass password1

NOTE: This will create a certificate that will expire in 3 months. You can use the –validity parameter setting it up to 9999 (for the equivalent of no expiration date).

2.4 Features with Limitations/Restrictions Removed for This Release

N/A

2.5 Features with Changes in This Release

N/A

2.6 Features with Dependencies on other Products

N/A

3 Installation and Upgrades

3.1 New Installations

With the first time (new user) installation of Orchestration Design 6.0, an image that has all of the pre-requisites and tools is being provided. In the cases where Eclipse or other pre-requisite components are already installed, incremental sets of plug-ins for the two tool sets are provided. For instructions to install Orchestration Designer, please refer to the Installation Guide which is provide as a stand-alone document.

3.2 Upgrades – General

There are framework updates in this build that may result in Java compilation errors after upgrading. To correct this, regenerate projects reporting errors.

Upgrades assume the user is starting from Dialog Designer 5.1, 5.0, 4.0 or 4.1.

3.2.1 AES (CTI) Applications with the Dial Operation

Upgraded applications that contain a CTI Dial operation will need a manual modification in the data node to add a value to the property: Caller Call Info Variable. Typically, this value is set to cticallinfo which is the default call info variable OD uses for caller CTI information. This is only applicable to if converting from DD 5.0.15 or earlier.

3.2.2 Remove Older Jar files after upgrade

After upgrading to OD 6.0 please make sure you do not have old versions of the following jar files on the application server:

commons-httpclient-3.1.jar
commons-logging-1.1.1.jar
log4j-1.2.15.jar
wss4j-1.5.8.jar

3.3 Upgrades with CTI

With the 6.0 release of AAOD, the cticonnector has changed names to the aesconnector. If you are executing a 5.1 application in a 6.0 environment, the new aesconnector is not compatible with your 5.1 application. You will need to copy the cticonnector.war (or .ear) file from your 5.1 environment and deploy it to the 6.0 environment. You may choose to codeploy the cticonnector and the aesconnector onto the same appserver. The 2 components can coexist, however this configuration is not recommended due to efficiency reasons and unless you are executing both 6.0 and 5.1 CTI enabled apps that need both connectors, it is recommended you remove or disable the aesconnector if you are just executing 5.1 applications.

If you are upgrading a 4.x application that uses CTI conferencing (add/remove conference party), you must manually edit the callflow file and explicitly select the active call ID attribute.

3.4 Upgrades from Dialog Designer 4.1 – CCXML

If you are using web services or database access with CCXML 5.0, you need to change the constructor:

```
public getSayings( com.avaya.sce.runtimecommon.SCESession mySession ) {

To

public getSayings( com.avaya.sce.runtimecommon.IRuntimeSession mySession ) {
```

3.5 Dialog Designer 4.0/4.1/5.0 Runtime

A new wizard (Avaya Speech Development -> Export Runtime Support Files) has been added and will produce the runtimeconfig web application and the runtime support ZIP files for the selected application server. The options to create these files no longer exist on the project export wizard pages.

3.6 Additional Application Installation Instructions for Websphere

- Extract the runtimesupportWebsphere.zip files to your classpath. We recommend copying them to your lib\ext directory.
  Example:
  `WS_HOME\AppServer\lib\ext`

- Once the app is running, to navigate to your index page for your speech application via a browser, you should include index.html within the URL as Websphere does not automatically default to this path.
  Example:
  http://myserver:port/mySpeechApp/index.html

- When deploying DDADMIN (a new component with 4.0) on WebSphere 6.1, you need to manually set jsp compilation to version 1.5 to get it to compile. At WebSphere Integrated Solution Console, do the following to install the DDADMIN ear file – runtimeconfig.ear

  1. Application > Install Application. At “Specify the EAR, WAR, JAR, or SAR module to upload and install,” window, Select “Local file System.” Browse to runtimeconfig.ear. Select “Show me all installation options and parameters.” Click Next.
  2. At “Choose to generate default bindings and mappings” window, click Next.
  3. At “Specifies the resulting security warnings from an analysis of this application” window, click “Continue.”

4. At “Select installation options” window, select “Precompile JavaServer Pages files” and keep the other default values. Click Next.
5. At “Map modules to servers”, select the runtimeconfig.ear. Click Next.
6. At “Provide options to compile JSPs” window, select “runtimeconfig.ear, WEB-INF/web.xml”, write “15” under “JDK Source Level.”
7. At “Provide JSP reloading options for Web modules” window, click Next.
8. At “Map shared libraries” window, click Next.
9. At “Map virtual hosts for Web modules” window, select “runtimeconfig.ear”, click Next.
10. At “Map context roots for Web modules” window, click Next
11. At “Summary” window, click Finish.
12. Make sure to click Save after installation. You can access DDADMIN at *http://myserver:port/runtimeconfig*

- When using Websphere 7 or above, these additional steps are required:

1. Run the osgiconsole.bat script from Websphere/appserver/bin.
2. In the command prompt, run "install file:///<websphere root
dir>/appserver/lib/ext/VPAppLogClientOSGI.jar"
*note: it has to be a valid URL path (for example, replace any space character with %20 in the path)
3. An id will be returned from step 2. then run "start <id>"
4. Copy the VPAppLogClientOSGI.jar into the Websphere/appserver/plugins directory
5. Restart Websphere.

3.7 Additional Application Installation Instructions for WebLogic

- Extract the runtimesupportWebLogic.zip files to your classpath. We recommend copying them to your domain server lib directory.
  Example:
  WL_HOME\samples\domains\wl_server\lib

- Using OD export to obtain a war file of your application, such as mySpeechApp.war, explode mySpeechApp.war to a folder which ends with ".war". For example, you can name this folder mySpeechApp.war also. Upload mySpeechApp.war folder to WebLogic, and choose "I will make the deployment accessible from the following location", where the location references the path to the mySpeechApp.war folder.

- Once the application is running, you can navigate to your index page for your speech application via a browser.

  Example:
  http://myserver:port/mySpeechApp

- To deploy your application war file non-exploded (not recommended) see Known Issues item #20.

3.8 Additional Application Installation Instructions for Tomcat

Tomcat 5.x:

- Extract the runtimesupportTomcat.zip files to your classpath. We recommend copying them to the Tomcat common directory.
  Example:
  TOMCAT_HOME\common

- Once the application is running, to navigate to your index page for your speech application via a browser, you can click on your application in the Tomcat manager site.
- A default “log4j.properties” file is placed in the TOMCAT_HOME/common/classes directory. You should examine this file to determine if it meets your logging needs. By default, the logging is set to “INFO”

which will generate a lot of log data in the TOMCAT_HOME/logs/Catalina.out file, which may grow quite large. You may consider changing the log level to WARN to reduce the amount of log output.

Example:
log4j.rootCategory=WARN, CONSOLE

Tomcat 6:

- Extract the runtimesupportTomcat6.zip files to your classpath. We recommend copying them to the Tomcat root directory.

  Example:
  TOMCAT_HOME

- Once the application is running, to navigate to your index page for your speech application via a browser, you can click on your application in the Tomcat manager site.
- A default “log4j.properties” file is placed in the TOMCAT_HOME/lib directory. You should examine this file to determine if it meets your logging needs. By default, the logging is set to “INFO” which will generate a lot of log data in the TOMCAT_HOME/logs/Catalina.out file, which may grow quite large. You may consider changing the log level to WARN to reduce the amount of log output.

  Example:
  log4j.rootCategory=WARN, CONSOLE

4 Known Issues

The following section describes any known issues, workarounds, words of caution, etc.

4.1 Limitations, Work Arounds, Notes:

AACC: CCMA Interoperability:

1. When using the integrated OD 6.0 tool suite and the Java 1.6 runtime, you may encounter errors caused by web service incompatibilities at the JRE level. To correct, copy the JAR files (geronimo-jaxws_2.2_spec-1.0.jar, jaxb-api-2.2.1.jar) to a subfolder under your Orchestration Designer Eclipse install and add to the eclipse.ini file under the vmargs: -Djava.endorsed.dirs=<path to folder with JAR files>
2. If you encounter errors connecting to the CCMA from Orchestration Designer for Contact Center application development, check that you have network access to the CCMA from your development machine. It may be necessary to configure proxy settings to bypass the proxy in order for OD to communicate with the CCMA.
3. If you are in the Contact Center perspective and have the Speech/Self-service call flow editor active, there will be two “View” menus. Close the self-service call flow editor or switch to the Speech perspective to remove the extra View menu.

CCXML:

4. In simulation, since there is only one input (microphone) and one output (speakers) for multiple call appearances, dialogs must be “serialized” to use the resources. Once a dialog starts a form, it must complete before the next form with a call appearance may use the resources. This generally provides a reasonable simulation of an application, however, in a CCXML application you have multiple call appearances and if one has input, a subsequent call appearance playing an informational message will be blocked until the input is completed and exists. This can occur in the ConferencePredefined application when a conference attendee hangs up. Since the incoming call has a dialog running, listening for an add or drop command, the message “XXX left the conference” will not be heard until the input on the incoming call completes. This does not happen on the platform.
5. When running certain applications (E.g. ConferencePredefined sample application), if the Dialog Designer->Avaya Application Simulator->Enable display of the Avaya Voice Browser output in a

console window is enabled, the application can behave irregularly. Please disable this if you encounter this behavior.

6. Applications that use the launchCCXML VP web service with DD/VP 4.0 will not work on VP 4.1 without modification. You now must include a send tag. For placement, one possibility is in the ccxml.loaded transition. The syntax for the addition is:

   ```
   <var name=”status”expr=”event$.reason”/>
             <send name=”’avaya.launchresponse’”
   targettype=”’avaya_platform’”target=”session.id”namelist=”status”/>
   ```

7. If the incoming call classification is “LIVE VOICE” then you need to go to the params tab in the simulation profile and select a new call classification then select “live voice”.
8. When working with the Sample Applications, InvokeDatabase and FindMe, there is a limitation when using a MSAccess database and multi-byte data. Multi-byte data will not be stored/retrieved properly. See http://forum.java.sun.com/thread.jspa?threadID=231861&messageID=3968823 for more details, but essentially, the default ODBC-JDBC driver does not handle UTF-8/16 correctly.
9. If there are any leading \r\n in a document before the ?,xml… the parse will fail. The work around in the JSP page is to have:

   ```
   <?xml version=”1.0” encoding=”UTF-8”?>
   <!DOCTYPE ccxml PUBLIC “http://www.w3.org/TR/ccxml” “ccxml.dtd”>
   <ccxml xmlns=“http://www.w3.org/2002/09/ccxml” version=”1.0”>
   ```

   for the first 3 lines before <%import...

CTI (AES):

10. When mapping extensions across 2 or more AES’s, the extensions can’t overlap.
11. The CTI (AES) Connector has a setting in the web.xml called providerstartupdelay (defaults to 4000 ms) that allows you to adjust a maximum delay value (in ms) to allow the provider to come in service. If the provider does not come into service before the timer expires, the connection is lost and will be retried.
12. For CTIC (AESC) deployment, tsapi.pro needs to be in common/lib on WebLogic.
13. For CTIC (AESC) deployment, tsapi.pro needs 2 additional properties for talking to AES 4.2 through secure connection. The avayaprca,jks comes with cticonnector.war. You just need to change the path to match your deployment. For example:

    ```
    trustStoreLocation=/opt/apps/cticonnector/WEB-INF/lib/avayaprca.jks
    trustStorePassword=password
    ```

14. You can now disable the application from gathering call information automatically by changing the flag sage.cti.autoinvoke.callinfo to false in your web descriptor. If you make this change, then you must use a CallInfo item in your data node and set the property: Initial Call to true before doing any CTI (AES) Call operations.
15. Starting with the JTAPI 3.1 driver, the OD CTI (AES) connector, can use the UCID (Universal Call ID) to keep track of call ID’s. To use this feature, you should enable it on your switch (consult your switch documentation). It is not mandatory to enable it, but it will guarantee no call ID’s are recycled which have been problematic in the past with high call volumes.
16. When passing data using the UUI, the maximum number of bytes you can use to write is 96. You can use a complex variable as the data for UUI. The properties allow entry of a variable/field in the dial operation.
17. When placing a call to a line on the same switch and it is in the busy state, you may get back a “failed” instead of a “busy” state indication. You should not check specifically for the “busy” state.
18. CTI (AES) does not support the “Numbered Services” configuration on the IR platform.
19. You can hear the status of the call if you are doing a blind call or a consult call with transfer on ring set. Note that you cannot hear busy or invalid line (fast busy) because the switch disallows a transfer to a busy line.
20. If you have CTI (AES) nodes in your application, you can’t run the application in HTML mode. The application will stop when it hits the CTI (AES) nodes.

21. The CTI (AES) Connector is configured to startup when the application server is started. You can disable this functionality if you do not have the CTI (AES) Connector configured yet or if you do not want it to run by editing the web.xml file found in your CTIConnector/WEB-INF directory on your appserver. Look for the value: <load-on-startup>1</load-on-startup> and remove or comment out this tag.

    NOTE: Setting it to 0 does not turn it off. You will need to manually start the CTIConnector if this flag has been removed or commented out.

Connectors:

22. It is not recommended to use the "Refresh" button on your HTML browser when an application relies on a connector (CTI, AES, IC, etc.). Due to the behavior of some HTML browsers, the refresh action causes the connectors to initialize a second time for the session which is not allowed. The recommended action is to close the HTML browser and then start up a new session.

Design Constraints:

23. The Orchestration Designer framework has constraints on the number iterations through call flow nodes that do not generate VXML. Those nodes are: Data, Servlet and Tracking. OD uses a servlet request forward to move between consecutive data nodes that do not generate VXML. This is faster and more efficient than redirecting the HTTP request. However, if you need to iterate more than 100 times on consecutive data nodes, you may run out of stack space. The recommendation if you need to iterate more than 100 times, is to move your iteration into Java code.

Deployment:

24. To deploy packed war files (vs expanded directory) on WebLogic, a directory is needed for writing application logs and recording files. By default, OD will create a directory with the application name in the OS's root ("C:/" for windows and "/" for Linux), and the log files and recording files will be written to the data/logs and data/temp directories respectively. Alternatively, you can set a specific directory by using the system property avaya.dd.temp.dir. This is where a sub-directory is created for each application. For example:

    -Davaya.dd.tempdir=c:\temp or some other value.

    In either case (default or tempdir), OD assumes that it can create directories and files in that location. If directory create is disabled, then you must create a subdirectory for each OD web application deployed with sub directories temp and data/log. When recording, temp recording files will be placed in temp and if tracing is enabled the trace.log and report.log will be in data/log. If you have a deployed application named Simple, under c:\temp you would see: c:\temp\Simple\temp and \data\log.

    If the avaya.dd.tempdir is not writable then recording will not work, nor will tracing and local reporting. The avaya.dd.tempdir can also be used with non-packed war files as well, thus providing a mechanism to move the temp and log files out of the web application to a separate area.

    Lastly when altering the location of the temporary directories,, the URL returned for the recorded file will be in the file:// protocol format and is NOT directly usable in a prompt to replay the recording since the file URL is not accessible on the platform where the voice browser is executing. To replay the message you will need to write java code to move the file from the temp location to a web context and reference the file there using the http protocol.

25. An OD application will fail to run if it is compiled with a Java version that is newer than the one on the Application Server. To fix this, select the Java Compiler compliance level that matches what is installed on the Application Server located at Window->Preferences->Java->Compiler in eclipse. For maximum compatibility, select 1.5/5.0 as the compiler compliance level.

26. To optimize garbage collection performance on the Application Server, we recommend adding the following:

    For Windows: Create a system level environment variable:

    ```
    export JAVA_OPTS = “-server -Xmx1024m -XX:MaxNewSize=30m -XX:+UseParNewGC -
    XX:+UseConcMarkSweepGC-XX:CMSInitiatingOccupancyFraction=60 –
    XX:ThreadStackSize=512”
    ```

    For Linux: when Tomcat is started as a service, add the flowing to the beginning of your /etc/init.d/tomcat file:

    ```
    JAVA_OPTS="-server -Xmx1024M -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -
    XX:ThreadStackSize=512"

      ulimit -n 8192
    ```

    NOTE: The Xmx1024M parameter we try to set to somewhere around half the size of the RAM configured on the box.
    Restart the machine after setting the JAVA_OPTS, and verify that the new setting are in effect by doing an: ps -ef | grep java

ECMA Script Expressions:

27. Single quotes that are part of the value of an ECMAScript expression will generate a semantic error in the Voice Browser if not properly formatted with an escape character (\). For instance:

    ```
    <var name="Output" expr="'John's return value'"/>    incorrect expression
    <var name="Output" expr="'John\'s return value'"/>   correct expression
    ```

    For Orchestration Designer to correctly generate the VXML output you need to add another escape character when building the expression in your callflow, otherwise Java will interpret it as an escape character itself.
    Within your call flow you would create the expression as follows:
    John\\'s return value

    This will insure the escape character is properly generated in the VXML output. Other characters to watch out for include ('gt;' for '>', '&lt;' for '<', and '&amp;' for '&').

Grammars:

28. Renaming a grammar column is now in the properties view. When the column is selected, the properties view will display the column name and column repeat. This was done to facilitate column repeats, and to better follow the Eclipse paradigm.
29. If you want to use SRGS/SISR with Nuance 9, select SRGS-SISR as the grammar format.
30. For external grammar deployment, the application web.xml must be edited to change localhost to the actual ip address of the application server. This can be done before or after deployment. As an alternative, you can use relative urls for external grammars MR 5381, or store the host and port in a OD variable and reference the OD varaible in the url %DDVAR_myextgrammarsloc%/Myapp/somedir/agrammar.grxml MR 5312
31. Built in speech grammars are provided by the ASR vendor. While most of them say they are supported, the specific grammars that are supported in any given language and the quality of the results when using them can vary. You should consult the speech vendor for specifics on the support for the languages you would like to use.
32. The quality of the Japanese and Chinese ASR provided by Microsoft (used in OD simulation) for built in grammars is not very high. When testing built-in grammars using Japanese or Chinese, it is recommended to use the “Send ASR” feature.

33. The following illustrates how the runtime can be used to generate large external grammars. The example is simple, but can be extended to build the grammar from an external data source. The reason to do this is for performance at runtime.

```
-----------------------------------------------------------------

import com.avaya.sce.runtime.DynamicGrammar;
import com.avaya.sce.runtime.GrammarItem;
import com.avaya.sce.runtime.GrammarRule;
import com.avaya.sce.runtimecommon.SCESession;

public class MyGrammar extends DynamicGrammar {

   public MyGrammar() {
   }

   public void buildGrammar(SCESession mySession) {
      GrammarRule rule;

      rule = new GrammarRule();
      rule.add(new GrammarItem("dog", "DOGTAG"));
      rule.add(new GrammarItem("cat", "CATTAG"));
      add(rule);
      rule = new GrammarRule();
      rule.add(new GrammarItem("blue", "BLUETAG"));
      rule.add(new GrammarItem("red", "REDTAG"));
      add(rule);
   }

   public static void main(String[] args) {
try {
    MyGrammarg = new MyGrammar();
    g.buildGrammar(null);
   String result = g.toString("en-us", SCERT.ASR_TYPES_MAP.get(SCERT.ASR_DISPLAY_LOGUENDO),
SCERT.VOICE_MODE);
               XmlWriter writer = new XmlWriter();
               Document doc = XML.getDocumentFromXMLString(result);
               writer.write(doc, System.out);
  } catch (Exception e) {
               e.printStackTrace();
  }
}
```

Help:

34. If you encounter the error "Runtime error" when using the on-line help, you may need to change you Internet Explorer settings to work properly with Sun's JavaDoc utility. While in your IE browser, go to Preferences, Tools, Internet Options. Select the Advanced tab. Make sure that "Disable Script Debugging (Internet Explorer)" and "Disable Script Debugging (other)" are checked. Click Ok. Close all IE windows and Eclipse. Then restart Eclipse.

IBM Grammars:

35. On the IR platform only, dynamic grammars with IBM ASR do not work properly. This is IR platform bug which requires a patch to the IR platform.

36. External Grammars get cached for 24 hours on the WVS. If you make a change to an external grammar, it will not take effect until the next day. This default can be modified as follows:

    - Browse to /opt/WebSphere/AppServer/bin
    - Run wsadmin.sh
    - After that run the following set of commands that display and set

    com.ibm.voice.server.rr.fetchexpires (which determines the cache expiration time in milsec) to desired value:
    set tmpCfg [$AdminControl queryNames *:*,type=WVSAdminConfig] (will retrieve system name and information)
    set cachetime [$AdminControl invoke $tmpCfg get com.ibm.voice.server.rr.fetchexpires] (will retrieve the current setting for cache expiration period)
    $AdminControl invoke $tmpCfg put {com.ibm.voice.server.rr.fetchexpires <cache expiration in msec}

IC:

37. To use SSL, you must provide your own certificate and either deploy to the Orchestration Designer Trust Store: trusted_weblm_certs.jks or your own external trust store which can be configured from the Orchestration Designer configuration servlet on the Certificates link. Similar configuration is necessary on the IC side. Please consult IC documentation for those details.
38. You cannot run an ICC enabled module in the simulator standalone as it will not invoke the ICC newcall operation. You must invoke it using a parent application that is also IC enabled.
39. If you implement a prompt to play while transferring a call to IC, the transfer can sometimes occur faster than the prompt is played. The flush prompt feature should be used for this scenario.
40. Some variables require a type. Specifically, variables used in a web operation might have to be converted to a specific type (int, long, etc). Fields on the VDU variable and VDU cache variable do not perform this conversion as they deal strictly with Strings. If you need to use a VDU field within your web operation that requires a type to be set, you should copy the VDU (or VDU cache) field into another local variable that does perform the conversion.
41. If you are running an IC-enabled application in the simulator, you must check to make sure there is a VOX or VRUSM/HTTPVox simulator configured in Window……IC Simulator.

Installaion:

42. If you are installing a fresh instance of eclipse 3.5 and are installing Orchestration Designer via the archive file without a name to the archive site location, after selecting the archive file, you might see Orchestration Designer disappear after a few seconds from the list of available software to install. This is a well known issue in eclipse 3.5.
    The workaround is to select the Orchestration Designer jar file from the "Work with:" dropdown and continue with the installation.

IPv6:

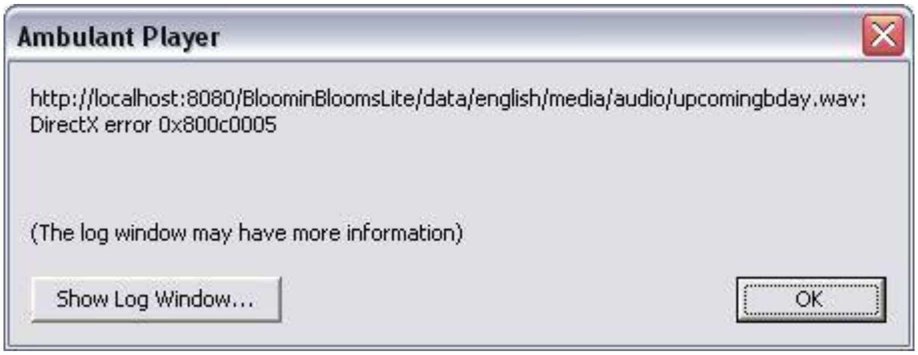
43. The MS SQL Server JDBC driver version 3.0 supports addressing using IPv6 format, however you must include the serverName connection string property in your connection URL for it to work. Please refer to JDBC driver documentation for additional details.

    Typical Connection URL format for IPv6 addressing:
    jdbc:sqlserver://;serverName=<ip address>;port=<port>;databaseName=MyDatabase

IVVR:

44. The simulator and media preview do not support RTSP.
45. If the text mode is set to “crawl” and the text writing mode is set to “rl” for a SMIL Text Block, the text is displayed in the Ambulant Player only if it doesn’t contain any spaces. This is a bug found in the Ambulant Player.

46. Although platform will play back 4 channel png files from your media page, the Ambulant player does not support them and will give you an error if you attempt to play them. You must use 3 channel png files in simulation.
47. Specify duration for the text, if the textmode is set to "crawl", else Ambulant player can hang.
48. If you upgrade to DD 5.1 or OD 6.0, the media preview view can be hidden and be next to the console view. Click on Window->Reset Perspective, so that the media preview view is placed next to the outline view.
49. When using OD via remote desktop, make sure the media preview view is big enough for the size of the video, otherwise the video is displayed outside the view.
50. If you see overlapping windows in the media preview view, make sure the view is closed on the perspectives other than the speech perspective.
51. If endsync is set to "media" in SMIL Par element, the OD app does not proceed to the next node in the simulator. This is a limitation of the Ambulant player.
52. If you have media items that have no intrinsic duration (text and images) you need to make sure they have an explicit duration set on their media properties. This is especially important during simulation which can cause unexpected results in the Ambulant Player and sometimes crashes.
53. If you are combining media items under a SMIL PAR element, pay attention to the end sync property and the durations (either implicit or explicit) of each media item to get your expected results. Likewise this applies to the VXML PAR when combining media pages with speech/phrases.
54. When using video images in parallel with other items, set endsync to “first” or specify a duration for the video image, otherwise the application may hang as it waits indefinitely for the image to stop playing.
55. If you see the following DirectX error from the Ambulant player when running IVVR apps, reset your proxy settings in Internet Explorer and add "localhost" to the ignore list.



Ambulant Player

http://localhost:8080/BloominBloomsLite/data/english/media/audio/upcomingbday.wav:
DirectX error 0x800c0005

(The log window may have more information)

Localization Packages/Certificate:

56. The certificates used to sign DD 5.0 were updated because they expired in August; consequently, localization packages that are installed with the previous certificate will produce warnings. New bundles with the current certificate will be made available with the GA of DD 5.0 and with subsequent service packs for 3.1, 4.0 and 4.1.

Memory Issues:

57. With the addition of the Eclipse WTP (Web Tool Project), memory issues have become more common. More information about the issues and workarounds can be found at the following links:

    Links that describe the issue:

    https://bugs.eclipse.org/bugs/show_bug.cgi?id=129490
    https://bugs.eclipse.org/bugs/show_bug.cgi?id=92250

    Similar threads on the Eclipse newsgroups:
    http://www.eclipse.org/search/search.cgi?cmd=Search!&form=extended&np=1&q=PermGen&tmptag=5&ul=/newslists/&wf=574a74

Possible workaround:
http://tassos.blogentis.net/2006/06/08/eclipse-and-permgen-space

Eclipse 3.4 appears to be even more memory intensive. Setting the permgen space to be 1024M seems to be a more stable setting.

Multi-Lingual:

58. Node (templates), Audio file names, grammar names, project variables, folders and project names can not contain double byte or extended ascii characters (E.g. Mandarin/Korean). This is because OD uses these names in urls/uris which must be composed of a subset of US ASCII characters. OD prevents the entry of unsupported characters.

OSDMs:

59. When using OSDMs with WebLogic 9.2, you will need to use the web.xml file to fix CT NSRD00019567 on OSDM-Core 2.0.4.2006062211. The patch should be obtained from the Nuance support site. To install the web.xml file on each WebLogic 9.2 platform:

    a. Copy your original osdm2-core.war (version 2.0.4) to osdm2-core.zip (for instance, in a shell window, type the following command: "cp osdm2-core.war osdm2-core.zip")
    b. Extract osdm2-core.zip into a folder named osdm2-core
    c. Extract the OSDM_Core_2.0.4_TS39109_CT19567_Patch.zip file to the osdm2-core folder. The file WEB-INF/web.xml will be updated.
    d. Log on to the WebLogic Server Administration console.
    e. From the "Domain Structure", click on "Deployments"
    f. Check that no osdm2-core application is already deployed, otherwise delete it.
    g. From the "Change Center", click on "Lock & Edit"
    h. The "Install" button from the "Deployments" tab should now be activated and selectable
    i. Click on "Install" in order to start deploying osdm2-core webapp
    j. Select the location of the osdm2-core folder just created in step 2 and click "Next"
    k. Select option "Install this deployment as an application" and click "Next"
    l. Click on "Finish" to finalize the deployment
    m. From the "Change Center", click on "Activate Changes"
    n. You should now see that the osdm2-core web application has been deployed and is in the "Prepared" state
    o. Check the box to the left of the title osdm2-core, click "Start" and click "Yes" to confirm. The osdm2-core state should now show "Active".

    If you want your OSDM to work with OSR 3.0, you need to edit the web.xml file to substitute osr 2.0 for osr 3.0 in the OSDM war (e.g. osdm2-core.war) to get the application to run successfully.

60. When using OSDMs with WebSphere 6.1, bad fetch errors occur with the core. There is a work around required. From Nuance:

    1) Log on to the WebSphere6 Administrative console
    2) Expand the applications option and click on "Enterprise Applications"
    3) You should be able to see the osdm2-core application already installed. Select the check box to the left of the title osdm2-core and then click "Stop" in the header menu. The screen should refresh and you should now see a red 'x' next to it.
    4) Log off from the WebSphere6 Administrative console
    5) Using a windows explorer, go to where the osdm2-core web application has been "installed" under the WebSphere6 application server folder such as the following path:

    'IBM\WebSphere\AppServer\profiles\<AppServerName>\installedApps\
    <MachineNodeCellName>\osdm2-core.ear\osdm2-core.war'

    6) Under this folder open/edit the following file: WEB-INF/web.xml

7) In the web.xml file, search for the following parameter name:

« com.speechworks.osdm.cookieVersion » (with no guillemets)

8) Change the value of this parameter (see tag <param-value>) to the following value:

« netscape » (with no guillemets)

9) Save and close the file
10) Using a windows explorer, go to where the osdm2-core web application has been “deployed” under the WebSphere6 application server folder such as the following path:

'IBM\WebSphere\AppServer\profiles\<AppServerName>\config\cells\
<MachineNodeCellName>\applications\osdm2-core.ear\deployments\osdm2-core\osdm2-core.war'

11) Under this folder open/edit the following file: WEB-INF/web.xml
12) In the web.xml file, search for the following parameter name:

« com.speechworks.osdm.cookieVersion » (with no guillemets)

13) Change the value of this parameter (see tag <param-value>) to the following value:

« netscape » (with no guillemets)

14) Save and close the file
15) Log on to the WebSphere6 Administrative console
16) Expand the applications option and click on "Enterprise Applications"
17) Check box to the left of the title osdm2-core and then click "Start" in the header menu. The screen should refresh and you should now see a green 'x' next to it.
18) Try the OSDM example "osdm2-core/en.us/examples/menu.jsp" and check that no more bad-fetch errors are reported in the VXML browser when fetching a page from OSDM core 2.x.

Simulation:

61. Running the same application twice in a row without restarting tomcat will generate exception errors. To work around this issue, enable “Automatically restart Tomcat when running an application”.
62. There is no support for G729 encoding in the simulator.
63. When simulating using the VRUSM simulator, it is recommended that you use the OD Runtime configuration servlet and set the ping timeout = 0 secs. This will effectively disable pings which aren't necessary for simulation. A future release/service pack will allow you to do this automatically via the simulation page in OD.
64. If you are experiencing slowness in simulation or unexpected delays, modify the garbage collection settings by adding either:

    -XX:+UseConcMarkSweepGC –XX:+UseParNewGC or –Xincgc to your eclipse startup.

    Example: Create a shortcut to eclipse and in the properties target:

    ```
    C:/SageRel-4.1\eclipse\eclipse.exe –vmargs –XX:+UseConcMarkSweepGC –XX:+UseParNewGC
    ```

    Additionally, add 2 entries in the Dialog Designer->Speech->Simulators table, one for vox->ic and one for ic->vox.

SSL:

65. When using SSL and Scansoft/Nuance ASR, grammars should be built as dynamic, even if they don’t need to be, so they will be processed, in-line. The ASR engine has a problem processing regular grammars if they exceed 4 elements.

Tomcat:

66. Tomcat 6.0.24 and newer added a new memory monitor, JreMemoryLeakPreventionListener, that may erroniously warn about memory leaks caused by threads started--but not stopped--by a web application. This is the expected behavior for Orchestration Designer applications caused by threads that are started by shared runtime (scertcommon.jar). These threads are shared by all Orchestration Designer applications and are not stopped intentionally.
67. Tomcat will reload the context of your speech application when it detects a change. This will occur pretty much anytime you change your project and save it while Tomcat creates a new current one for your servlet and reloads the dependencies. OD does not have a way to indicate to Tomcat that a submodule is a dependency of its parent application. Therefore, when the parent's context is loaded into the new class loader there is a conflict with the class loader that was used to load the submodule. The solution is two fold:
    a. If your context file points to your speech application, you can simply reload the context of the submodule right in OD using the Tomcat context option "Reload this context."
    b. If you have deployed your submodule into Tomcat and Tomcat is running it out of the webapps directory, the simplest solution is to restart Tomcat or you can use the Tomcat Manager application. Refer to the Tomcat documentation for more information on how to do this using the Tomcat manager.
68. The console window may display warning messages about invalid log4j configuration when starting Tomcat from Eclipse. To fix this add a log4j.properties file to the Tomcat classpath (i.e. to <tomcat home>/common/classes). The properties file should contain these lines:

```
log4j.rootLogger=info, stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{dd/MM/yyyy HH:mm:ss} %5p - %m%n
```

Transfers:

69. Bridged transfers require CM version 3.0.1 or above, with an H323 configurations on IR due to a Configuration Manager defect in earlier releases.

TTS:

70. WVS 5.1.3 GA version of IBM TTS has a bug in it that impacts the sound quality of TTS prompts. A fix has been posted to the IBM support site: PMR# 05274,756,000.
71. Using symbols in TTS (i.e. &, <, >, etc.) in a prompt may result in runtime errors. These symbols generate invalid VXML code. **Workaround**:
    a. Spell out the word. For example: '&'→"and", '<'→"less than", etc.

Validate Menu Option:

72. Running the validate menu option on any project causes a Warning to show up in the Problems view. The Resource is web.xml and the Location is line 8 which is: -><! DOCTYPE web-app PUBLIC"-//Sun Microsystems, Inc//DTD Web Application 2.3//EN" http://java.sun.com/dtd/web-app_2_3.dtd> There is currently no work-around to get rid of this warning once it occurs.

WebLogic Application Server:

73. When running on WebLogic 10.3.1 or above, the app losses control of the http session it starts initially. The http session would go idle, eventually times out and get removed by the application server. The time-out interval depends on what you set in the deployment descriptor either via the console or editing the weblogic.xml. If you are seeing this symptom, it is caused by WebLogic starting to have the cookie-http-only parameter set to true by default. You need to change the value of the parameter to false. To do that, you would need to deploy a weblogic.xml with the following content in the application's WEB-INF directory. If you already have the weblogic.xml, you would need to incorportate the following content. However if you think it's too much work, you can choose

not to do anything. We believe that the accumulation of http sessions (before they get removed by the system) should not be harmful since each http session used by OD is really light weight - we don't use it to store information.

weblogic.xml:

```
<!DOCTYPE weblogic-web-app PUBLIC "-//BEA Systems, Inc.//DTD Web Application 8.1//EN"
"weblogic810-web-jar.dtd">
<weblogic-web-app>
  <session-descriptor>
    <cookie-http-only>false</cookie-http-only>
  </session-descriptor>
</weblogic-web-app>
```

Web Services:

74. After generating a new web service operation, you may see warnings in the java code generated by Axis. These warnings are not harmful and you do not have to take any action to correct them.
75. The graphical Web Services tool is intended for use with simpler web services. I.e. Web services that return base java types (string, int). If your Web service returns an object that has nested objects, at a minimum, you will need to select "Use Java Obj" on the Web Service wizard. This will instruct the runtime code to take the returned value and place it into the Orchestration Designer variable as an Object. You will need to write some Java code to examine your returned Object and store values into Orchestration Designer variables. This is necessary as Orchestration Designer only supports a simple variable system of either a variable or a grouping of variables.

WebSphere Application Server:

76. If the application executes the number of database operations that exceeds the number of the connection pool specified for the data source, the trace log will say javax.servlet.ServletException: Connection not available, Timed out waiting for 180001. The database operation will hang for a while before the error is thrown. To fix this, the configuration should be changed as follows:
    - Add the following node to the web.xml of the application
        - <resource-ref>
        - <!— replace DS with the real data source name in OD ->
            - <res-ref-name>jdbc/DS</res-ref-name>
            - <res-type>javax.sql.DataSource</res-type>
            - <res-auth>Container</res-auth>
            - <res-sharing-scope>Unshareable</res-sharing-scope>
        - </resource-ref>
    - Export the app for deployment; make sure this is done after the web.xml has been changed. Otherwise, you have to re-deploy the app.
    - In the Websphere admin console, when you install the application, there will be a new step call Map resource references to resources for mapping the data source name in web.xml to the data source you normally define in Websphere admin console.
    - On the screen for this step, all the way at the bottom, find the Browser button, click on it, and select the data source that you've defined for the intended database.
    - Finish the deployment.

77. See the Database section for more information on database configuration for WAS.
78. If you have trouble deploying the Tsapi.pro file or WAS install scripts on a Linux machine, try using the utility dos2unix on the files.
79. Some database drivers work fine in Orchestration Designer when designing the speech application, but do not work in some versions of Websphere. In that case, replace the driver in Websphere with an updated or different version depending on your database type and configure the settings appropriately referring to the WebSphere documentation.

80. For the Help Topic: Installing project files on WebSphere servers, Item 4, should read:

Set this variable to point to your …\Websphere\appserver\bin directory.

5 Version Information

To view the Orchestration Designer version in the product, select Help, then About Eclipse, select Feature Details and select the Self Service entry.

6 Hardware Prerequisites

- CPU speed 1GHz (min) 2GHz (recommended)
- RAM 512 MB (min) 1GB (recommended)
- Hard Disk drive 40 GB (min)

7 Software Prerequisites (Supported Versions)

7.1 Design Environment

- Windows XP Professional
- Vista Ultimate
- Vista Business
- Windows 7 Ultimate
- Windows 7 Professional
- Eclipse 3.5 or 3.6
- GEF 3.5 or 3.6
- Tomcat version 6.0 or 5.5, **Note**: **Design & Runtime must use the same version of Tomcat or there will be issues with database connections.**
- J2SE version 1.6.x or J2SE version 1.5.x
- Microsoft Speech SDK 5.1 or 6.0
- Apache Axis 1.4
- Apache Axis 2.0
- DTP 1.8 (with Eclipse 3.5) or DTP 1.9 (Eclipse 3.6)
- EMF-SDO-XSD 2.5 (with Eclipse 3.5) or EMF-SDO-XSD 2.46 (with Eclipse 3.6)
- WTP 3.1 (with Eclipse 3.5) or WTP 3.2 (with Eclipse 3.6).

7.2 Runtime Environment

Not all of the components are required for the runtime (E.g. the application server will be either Tomcat; WebSphere or Oracle WebLogic, the OS will be Windows, Solaris or Linux; the platform will be either IR or VP, etc.). Supported versions are listed, but the vendors supported list of dependencies and combinations should be consulted when setting up the environment). The tested configurations are as listed in the requirements document.

- Windows Server 2003 standard R2 and later
- Windows Server 2008 R2 and later
- Linux ES ES 5.0, ES 6.0
- IBM AIX 6.1
- Tomcat 6.0 or 5.5, **Note: The design and runtime version of Tomcat must match or there will be issues with database connections.**
- J2SE version 1.6.x or J2SE version 1.5.x
- Apache Axis 1.4
- Apache Axis 2.0
- Nuance OSR 3.0.10
- Nuance (aka Quantum) 9.0
- Loquendo 7.5.35
- IR 4.0

- Voice Portal 5.1
- Avaya Aura® Experience Portal 6.0
- WebLM Version 4.5
- Websphere 6.1 application server
- Websphere 7.0 application server
- Websphere Voice Server version 5.1.3
- Oracle Weblogic 10, 10.3 and 11g
- Sun Solaris 10 on SPARC
- CM version R013x01.2.639.0 or greater, and must have the green feature 7434ND enabled and licensed is required to use the Consultation transfer feature.