



**Dialog Designer
(DD)
Release 4.1.10
Release Letter
(Supersedes All Previous Releases)**

Date: February 2, 2009

Table of Contents

1	Release Information.....	1
2	Features	2
2.1	MRS INCLUDED IN THIS RELEASE	2
3	Installation and Upgrades	3
3.1	NEW INSTALLATIONS	3
3.2	UPGRADES FROM DIALOG DESIGNER 3.0	3
3.3	UPGRADES FROM DIALOG DESIGNER 3.1	4
3.4	UPGRADES FROM DIALOG DESIGNER 4.0 – NO CCXML	4
3.5	UPGRADES FROM DIALOG DESIGNER 4.0 – CCXML APPLICATIONS	4
	<i>DTD Changes:</i>	4
	<i>Event Changes:</i>	4
	<i>Object Changes:</i>	4
3.6	DIALOG DESIGNER 4.0/4.1 RUNTIME	5
3.6.1	<i>Co-resident 3.1, 4.0 and 4.1 Applications</i>	5
3.7	ADDITIONAL APPLICATION INSTALLATION INSTRUCTIONS FOR WEBSHERE.....	6
3.8	ADDITIONAL APPLICATION INSTALLATION INSTRUCTIONS FOR WEBLOGIC	6
3.9	ADDITIONAL APPLICATION INSTALLATION INSTRUCTIONS FOR TOMCAT	6
4	Known Issues.....	7
4.1	LIMITATIONS, WORK AROUNDS, NOTES:	7
5	Version Information	15
6	Hardware Prerequisites	15
7	Software Prerequisites (Supported Versions)	16
7.1	DESIGN ENVIRONMENT.....	16
7.2	RUNTIME ENVIRONMENT.....	16

1 Release Information

This release note describes build 4.1.10 of Dialog Designer. This is the third service pack to the GA 4.1 release and contains MR bug fixes. The primary features of the 4.1 release are:

- Usability
 - Multiple Call Flows
 - Expanded Web Services support

- CCXML Conferencing and Merge features (phase 2)
- Currency with 3rd party products
 - Support for MicroSoft Vista OS in the development environment
 - Eclipse 3.3 – required for Vista
 - Upgrade to latest CCXML specification

2 Features

This section describes the status of the features that were developed for this release.

2.1 MRs included in this Release

id	severity	component	summary
wi00241536	High	Call Flow Editor	NPE in Validator keeps you from making any changes to Project
wi00240331	High	CTI Connector	IR3 SP3 - calltag changed
wi00247178	Med	App Config UI	Configurable Application Variables can't be retrieved with proxy settings configured
wi00251540	Med	App Config UI	ddadmin fails to load ddconfig.xml file if path contains spaces
wi00278010	Med	App Config UI	Error messages when starting tomcat
wi00120525	Med	Call Flow Editor	no error marker for subflow w/out return
wi00245522	Med	CCXML Connectors	Input vars not generated for DBOPs that use WHERE clause in CCXML code
wi00121694	Med	CCXML Runtime/Simulation	when outcall answered with recoded_msg need to ariable msg_end
wi00276681	Med	Deployment	proxy is set wrong in validate.jsp
wi00120013	Med	Grammar Editor	if first slot is optionl interpretation is wrong.
Wi00230536	Med	IC Connector	IndexOutOfBoundsException execption if VOX response message has bad format
wi00230819	Med	IC Connector	VOX socket read error if less than 1024
wi00245425	Med	IC Connector	DD 4.1 ICCWeb deployment for Websphere app.xml wrong war file
wi00230807	Med	JDBC Connector	Cursor remains open after stored procedure is called
wi00241237	Med	JDBC Connector	incorrect resource attributes generated in context file
wi00123687	Med	Licensing	App server must be restarted if unable to get initial license
wi00231368	Med	Localization Packages	number formatting problems with zh-cn & zh-hk
wi00111424	Med	Runtime Framework	Need alternate URL encoding for ext grammar URLs
wi00230891	Med	Runtime Framework	time:hour variable should use 0-23 format rather than 1-24
wi00236208	Med	Runtime Framework	if ariable and backup are in the same collection, iteration is double incremented
wi00281376	Med	Runtime Framework	Using variables in thresholds on goto cause compile err

wi00231436	Med	Simulation	simulation scripts not processed
wi00243598	Med	Web Services Connector	Log error when generating web service stub
wi00113845	Low	App Config UI	allow leading 0 on tserver extensions.
wi00112265	Low	CCXML	Change to undo wi00074306 RND0100005891
wi00232659	Low	IC Connector	set the vducache id field when the id is set after a newcall
wi00075491	Low	Runtime Framework	abort a request on timeout.
wi00244562	Low	Runtime Framework	change runtime to use only ont timer thread.
wi00120751	Low	Variable Editor	entering a " in a variables value causes xml problems.

3 Installation and Upgrades

NOTE: If you open the preferences and set tomcat home before activating a DD plugin (I.e. before switching to the speech perspective or launching a new project wizard, etc.), then DD doesn't know when Tomcat home is set, so it doesn't copy all of the jars or runtime config webapp. If you switch perspective, or do something that triggers the initialization of the DD plug-ins, then everything works fine. In some cases, this does not work and it is necessary to set Tomcat to an invalid directory, and then reset it back to the correct directory while in the speech perspective to correct the problem.

3.1 New Installations

With the first time (new user) installation of Dialog Designer 4.1, the pre-requisite components such as Eclipse, Tomcat and Java will be included on the media with their own installers for convenience, but the user must install them separately before installing Dialog Designer. The CCXML components require additional training and a password to install and enable them.

For instructions to install Dialog Designer 4.1, please refer to the documentation by Information Solutions included in the on-line documentation. A stand alone pre-requisites document is available for download prior to installing DD.

3.2 Upgrades from Dialog Designer 3.0

Dialog Designer 4.1 uses a new version of Eclipse (3.2 or 3.3) which is not backward compatible with plug-ins built for Eclipse 3.0 (DD 3.0 plug-ins). For customers moving from DD 3.0, with applications that they would like to move forward to DD 4.1, we recommend installing a separate instance of the DD environment (Eclipse, Tomcat, Java, etc.) and maintaining a copy of the previous version of 3.0. We also recommend that users maintain separate workspaces for their different DD versions. For any 3.0 projects that they want to update to 4.0, they should copy (using Windows Explorer) the project into the 4.0 workspace and import into DD 4.1. Projects updated to 4.1 cannot be opened again with DD 3.0.

The upgrade process for applications built on DD 3.0 is simple, but does require regenerating applications after importing them before running them in simulation mode. It does not require code changes with a couple exceptions. The new version of Axis (1.4) handles arrays differently than the version of Axis used with Dialog Designer 3.0 (Axis 1.2). If you developed custom java code to handle arrays for Web Services used in a Dialog Designer 3.0 application, you will need to make some edits to be compatible with the new version of Axis. Also, new project variables are automatically added to the project to support menus and slots. In the event that there are name conflicts or incompatible variable types, the upgrade process will fail and users must manually resolve the conflicts. Any conflicts are logged in the Error Log.

NOTE: The framework API changed so that when you generate a project, 4.1 will generate the proper code for all files that are managed by the project, but any orphaned java classes are no longer managed by the project and therefore are not regenerated, and will create an error. You must remove the orphaned files to get rid of the error.

There are new features that require some edits to use the new features. E.g., to use multiple CTI servers, there are edits required for configuration. These are described in detail in the documentation.

For general instructions to upgrade to DD 4.1, please refer to the documentation by Information Development.

3.3 Upgrades from Dialog Designer 3.1

There are no unique upgrade steps when upgrading the development environment from Dialog Designer 3.1 to 4.1. It is recommended that you save backup copies of your 3.1 applications before the upgrade. Also, it is recommended that projects are copied (using Windows Explorer) into the 4.1 workspace, then imported into DD 4.1.

3.4 Upgrades from Dialog Designer 4.0 – no CCXML

There are no unique upgrade steps when upgrading the development environment from Dialog Designer 4.0 to 4.1 for applications that do not use CCXML. It is recommended that you save backup copies of your 4.0 applications before the upgrade. Also, it is recommended that projects are copied (using Windows Explorer) into the 4.1 workspace, then imported into DD 4.1.

3.5 Upgrades from Dialog Designer 4.0 – CCXML Applications

Dialog Designer 4.1 (and VP 4.1) now uses the latest revision of the CCXML specification. The latest version introduces some backward compatibility issues that require changes to your CCXML applications built with 4.0. This section details the changes that impact applications. These changes will be formally delivered in the phase 2 DD release of 4.1, but the changes to the applications are being communicated in advance so that the upgrade process can precede the newer version.

DTD Changes:

send	Data attribute changed to name.
dialogprepare & dialogstart	Added parameters attribute.
	Parameters go to the vxml dialog and namelist goes to the application server.
	Review every namelist in dialogprepare and dialogstart. Recommend changing namelist to parameters or adding parameters.
transition	Removed the name attribute, events are referenced as event\$
	Change evt to event\$
	Remove name="evt"
	Change (ext) to (event\$)

Event Changes:

Removed	error.dialog.wrongstate
	error.merge
Added	connection.merge.fail
	connection.redirect.fail
	connection.accept.fail
	connection.reject.fail

Object Changes:

Dialog Object	Added input, outputs (affects join/union)
	Removed duplex, endpoint

Conference Object	Added conferencename
Connection Object	Added inputs, outputs (affects join/union)
	Removed endpoint, duplex

3.6 Dialog Designer 4.0/4.1 Runtime

There is a new component that is required to run 4.0/4.1 applications. The deployment wizard offers a check box for the new Administration component. You will need to check this for the first 4.1 application you wish to deploy. A new runtimeconfig.war file needs to be copied to the application server deployment directory and installed.

The runtime files: runtimeSupportWebsphere, runtimeSupportWeblogic and runtimeSupportTomcat are optionally created during the project export step. The appropriate file must be installed on the application server before running a 4.1 application.

There are also some files that are out of date and need to be removed if you are upgrading an existing DD runtime application server. The following details this list of files.

commons-logging.jar (NOTE: only remove this file on Tomcat or Websphere configurations)
commons-logging-1.0.4.jar
scertcommon-*.jar
VAppLogClient.jar
VAppLogClient_4.0.0.jar
wsdl4j-1.5.1.jar

3.6.1 Co-resident 3.1, 4.0 and 4.1 Applications

Applications built with Dialog Designer 4.1 can share the same application server as applications built with DD 3.1 and 4.0. The 4.1 framework is backward compatible, however in the case of 3.1 applications that use the CTI or IC connector; they must use the new 4.0 or 4.1 versions of the connectors. This means that steps must be taken to ensure these connectors are installed with the 4.1 runtime. The 3.1 or 4.0 framework must be removed and replaced by the 4.1 framework, but 3.1 and 4.0 applications will continue to work with the 4.1 framework. It is not backward compatible with 3.0 applications.

DD ADE Version used to build App	Application Runtime Version	Supported	Notes/Comments
3.0	3.0	x	
	3.1		Requires Java 1.5
	4.0		Requires Java 1.5
	4.1		Requires Java 1.5
3.1	3.0		Won't run with Java 1.5
	3.1	x	
	4.0	x	Must use Tomcat 5.0
	4.1	x	Must use Tomcat 5.0.
4.0	3.0		Won't run with Java 1.5
	3.1		
	4.0	x	
	4.1	x	CCXML applications must be upgraded
4.1	3.0		Won't run with Java 1.5
	3.1	x	
	4.0	x	
	4.1	x	

If using Tomcat, Tomcat version 5.0 must be used since 3.1 applications will not work with Tomcat 5.5.

3.7 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>

3.8 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 DD 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>

3.9 Additional Application Installation Instructions for Tomcat

- 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

4 Known Issues

The following section describes any known issues, workarounds, words of caution, etc, and a table of outstanding bug MRs for DD.

4.1 Limitations, Work Arounds, Notes:

Caching:

1. Caching has been disabled in Dialog Designer. The documentation related to this should be ignored as setting items to be cached is not currently supported.

CCXML:

2. 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.
3. 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.
4. 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"/>
```

5. If the incoming call classification is “LIVE VOICE” then you need to go to the params tab in the simulator and select a new call classification then select “live voice”.
6. 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.
7. 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:

8. 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 Call operations.
9. The JTAPI 3.1 driver included in this release of the DD CTI 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.
10. In the 4.0 CTI documentation, under the CTI tab, Appendix B, only step 1 is valid. Please refer to Appendix G "Configuring Dialog Designer CTI Applications to work with Avaya IVR Systems" for the latest info for step 2.
11. When passing data using the UI, the maximum number of bytes you can use to write is 96. You can use a complex variable as the data for UI. The properties allow entry of a variable/field in the dial operation.
12. 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.
13. CTI does not support the "Numbered Services" configuration on the IR platform.
14. 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.
15. If you have CTI nodes in your application, you can't run the application in HTML mode. The application will stop when it hits the CTI nodes.
16. In appendix G of the Developer Guide, within the section entitled 'Constructing a call and data transfer using CTI connectors', in the third bullet item of step 1.f, the last part of the sentence should read 'use the **state** field'. Currently, it says 'use the **callid**'.
17. Support for transfers with and without CTI will vary depending on the configuration, platform and software version the platform is running. Please refer to the matrix created for IR1dot3 Support.xls file on the DD CD for a matrix of supported features. For other Voice Portal and IR 2.0 release, this information will be included in the platform documentation.
18. If the TServer goes down, the JTAPI provider has some renegade threads which can only be stopped by the TServer recovering and coming back up, or, manually killing Tomcat. A normal shutdown of Tomcat will not get rid of the threads because they are running.
19. The CTI Connector is configured to startup when the application server is started. You can disable this functionality if you do not have the CTI 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.

Design Constraints:

20. The Dialog Designer framework has constraints on the number iterations through call flow nodes that do not generate VXML. Those nodes are: `Data`, `Servlet` and `Tracking`. DD 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:

21. The on-line documentation in DD for using WebLogic is incorrect. Use the information in this document in section 3.6.
22. To optimize performance on the Application Server, we recommend adding the following:

For Windows: Create a system level environment variable:

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

```
export JAVA_OPTS
```

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:

23. 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 Dialog 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 ('>' for '>', '<' for '<', and '&' for '&').

Grammars:

24. 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 DD variable and reference the DD variable in the url
%DDVAR_myextgrammarsloc%/Myapp/somedir/agrammar.grxml MR 5312
25. Built in grammars must be built by hand for use on IR (QQ items 172406 – IR 1.3 and 172408 – IR 2.0). IR current expects the language to be sent as a query string, however per the VoiceXML specification, language is handled through the xml:lang attribute, which is how DD will generate them. This works on Voice Portal.
26. 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.

27. The quality of the Japanese and Chinese ASR provided by Microsoft (used in DD 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.
28. 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) {  
        MyGrammar g = new MyGrammar();  
        g.buildGrammar(null);  
        System.out.println("<?xml version='1.0' encoding='UTF-8'>");  
        System.out.println(g.toString("en-us", com.avaya.sce.runtimecommon.SCERT.ASR_IBM,  
            com.avaya.sce.runtimecommon.SCERT.VOICE_MODE));  
    }  
  
}
```

Help:

29. If you encounter the error “Runtime error” when using the on-line help, you may need to change your 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:

30. 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.
31. 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:

32. 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.

Memory Issues:

33. 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=/newlists/&wf=574a74>

Possible workaround:

<http://tassos.blogentis.net/2006/06/08/eclipse-and-permgen-space>

Menu Variables:

34. There is a bug in the IR and VP AVB code so that the interpretation and value fields of a Menu variable are not set with any value after ASR or DTMF recognition for a menu choice. This is fixed in Voice Portal patch 3.0.1.2.

Multi-Lingual:

35. 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 DD uses these names in urls/uris which must be composed of a subset of US ASCII characters. DD prevents the entry of unsupported characters.

Nuance 9.0 and NBest:

With Nuance 9.0, a capability was added to distinguish between phonetic differences (cat, hat, bat) and natural language differences (bugg, bug). When a phonetic difference is encountered by the Nuance 9 recognizer it is returned as a unique result in application.lastresult[n]. Example:

```
application.lastresult[0].interpretation = cat
application.lastresult[1].interpretation = hat
application.lastresult[2].interpretation = bat
```

When a natural language result is encountered by the Nuance 9 recognizer it is returned as multiple interpretations within a result set `application.lastresult[n].interpretation[i]`. Example:

```
application.lastresult[0].interpretation[0] = bug
application.lastresult[0].interpretation[1] = bugg
```

This additional nesting is not supported in Dialog Designer. The result is the application will only see the first interpretation (bug). In order to force nuance to report all differences as unique result sets, the `NSSserver.cfg` file needs to be altered by adding `nlsminterpretation=1`.

#The OSR result format

```
server.mrcp2.osrspeechrecog.mrcpdefaults.VSP.server.osrspeechrecog.result.mediatype VXIString
```

```
application/x-
```

```
vnd.speechworks.emma+xml;strictconfidencelevel=1;mrcpv=2.06;nlsminterpretation=1
```

#The OSR result format

```
server.mrcp1.osrspeechrecog.result.mediatype VXIString
```

```
application/x-vnd.speechworks.emma+xml;strictconfidencelevel=1;nlsminterpretation=1
```

After making this change, the results will be returned as:

```
application.lastresult[0].interpretation = bug
application.lastresult[1].interpretation = bugg
```

and Dialog Designer will map this as expected to variables.

OSDMs:

36. Pre OSDM 2.0.6 versions will not work with the new Nuance 9.0 (Quantum release). The new OSDM versions have not as yet been validated with DD 4.0.
37. 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.

38. 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.

Runtime Config:

39. In Dialog Designer 4.0, there is a new application administration web application that must be deployed on the application server. This web application is used to configure application server parameters like the license server url, CTI and IC settings, etc. On the desktop (simulation), this application is automatically deployed on your local Tomcat so that the desktop runtime simulation

has access to configuration data. However, on the desktop you MUST enter the configuration values using the preference pages and NOT the configuration web application.

In general, you will not even know that Dialog Designer has deployed the configuration web application on the desktop unless you enter the Tomcat Manager. In summary, in the development environment, use the preference pages to configure application values. When deploying an application to the application server, you must also deploy the runtimeconfig.war file and use this application to configure application parameters.

Simulation:

40. 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:

41. 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:

42. 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 sevlet and reloads the dependencies. DD 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 DD 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.
43. 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:

44. 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:

45. 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.
46. 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:

47. Running the validate menu option on any DD 4.0 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.

Web Services:

48. 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 Dialog Designer variable as an Object. You will need to write some Java code to examine your returned Object and store values into Dialog Designer variables. This is necessary as Dialog Designer only supports a simple variable system of either a variable or a grouping of variables.
49. When using Web Services, operation name overloading in a wsdl:portType is disallowed by the Profile. A wsdl:portType in a description must have operations with distinct values for their name attributes. Note that this requirement applies only to the wsdl:operations within a given wsdl_portType. A wsdl:portType may have wsdl:operations with names that are the same as those found in other wsdl:portTypes.

WebSphere Application Server:

50. See the Database section for more information on database configuration for WAS.
51. 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.
52. Some database drivers work fine in Dialog 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.
53. 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.

WebLM:

54. If you are using a Solaris machine for your WebLM server, you must use WebLM version 4.1, or later or the host id provided in the RFA generated license will not work.

5 Version Information

To view the Dialog Designer version in the product, select Help, then About Eclipse, select Feature Details and select the Dialog Designer 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
- Vista Ultimate
- Vista Business
- Eclipse 3.2.2 or 3.3
- GEF 3.2.2 or 3.3
- Tomcat version 5.5 (preferred) *or* Tomcat 5.0.28, **Note: Design & Runtime must use the same version of Tomcat or there will be issues with database connections.**
- J2SE version 1.5.x
- Microsoft Speech SDK 5.1
- Apache Axis 1.4

For CCXML:

- DTP 1.5 (with Eclipse 3.3)
- JEM-runtime-1.2.3_jem
- EMF-sdo-runtime-2.2.2
- EMF-SDO-XSD 2.3 (with Eclipse 3.3)
- WTP-R-1.5.3 (WTP 2.0 with Eclipse 3.3)
- Xsd-runtime-2.2.2

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 BEA 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
- Linux ES 4.0
- Tomcat 5.5.x (preferred) *or* Tomcat 5.0.28, **Note: The design and runtime version of Tomcat must match or there will be issues with database connections.**
- J2SE version 1.5.x
- Apache Axis 1.4
- Nuance OSR 3.0.10
- Nuance 8.5
- Nuance (aka Quantum) 9.0
- IR 1.3 or IR 2.0 or IR 3.0
- Voice Portal 3.0.1.x, Voice Portal 4.0, build 15 or later and VP 4.1
- WebLM Version 4.5
- Websphere 6.1 application server
- Websphere Voice Server version 5.1.3
- BEA Weblogic 9.1, 9.2 is required if using OSDMs
- 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.