Dialogic® Vision™ VoiceXML Administration Manual

June 2009 64-0401-01

www.dialogic.com

Copyright and legal notice

Copyright © 2005-2009 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation or its subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Any use case(s) shown and/or described herein represent one or more examples of the various ways, scenarios or environments in which Dialogic® products can be used. Such use case(s) are non-limiting and do not represent recommendations of Dialogic as to whether or how to use Dialogic products.

Dialogic, Dialogic Pro, Brooktrout, Diva, Cantata, SnowShore, Eicon, Eicon Networks, NMS Communications, NMS (stylized), Eiconcard, SIPcontrol, Diva ISDN, TruFax, Exnet, EXS, SwitchKit, N20, Making Innovation Thrive, Connecting to Growth, Video is the New Voice, Fusion, Vision, PacketMedia, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation or its subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 9800 Cavendish Blvd., 5th Floor, Montreal, Quebec, Canada H4M 2V9. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Revision history

P/N 64-0401-01

Revision	Release date	Notes
A	June 2009	DEH/BK, Dialogic® Vision™ VX Integrated Media Platform 4.1
Last modified: June 8, 2009		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

Table Of Contents

1 Introduction

The *Dialogic® Vision™ VoiceXML Administration Manual* provides detailed information about configuring the Vision™ VoiceXML Subsystem in the Dialogic® Vision™ VX Integrated Media Platform. The material in this manual supplements the material provided in the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

This manual assumes that you are familiar with the VoiceXML standard. It also assumes that you have read the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*, which describes how to configure and develop applications for the Vision™ VX Integrated Media Platform.

Note: The product to which this document pertains is part of the NMS Communications Platforms business that was sold by NMS Communications Corporation (“NMS”) to Dialogic Corporation (“Dialogic”) on December 8, 2008. Accordingly, certain terminology relating to the product has been changed. Below is a table indicating both terminology that was formerly associated with the product, as well as the new terminology by which the product is now known.

Former terminology	Current terminology
Vision VoiceXML Server	Dialogic® Vision™ VX Integrated Media Platform

2 Overview

Overview of the VoiceXML Subsystem

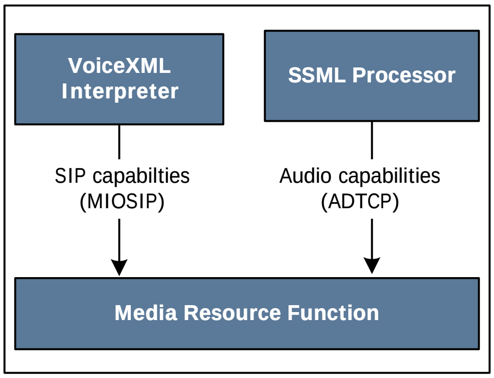
The VoiceXML Subsystem is the subsystem of the VX Integrated Media Platform that interprets VoiceXML dialogs and provides media processing for VoiceXML applications.

Components

The VoiceXML Subsystem has three components:

Component	Description
VoiceXML Interpreter	Interprets VoiceXML dialogs. At startup, the VoiceXML Interpreter loads the Media Interface Object (MIOSIP), which provides SIP capabilities for the Media Resource Function. For information about configuring the VoiceXML Interpreter, see *Overview of configuring the VoiceXML Interpreter* on page 11.
SSML Processor	Processes SSML requests, including requests for third-party text-to-speech (TTS) resources. At startup, the SSML Processor loads the Audio Driver (ADTCP), which provides audio capabilities for the Media Resource Function. For information about configuring the SSML Processor, see *Overview of configuring the SSML Processor* on page 31.
Media Resource Function	Provides media processing including record and playback, and provides interfaces to automatic speech recognition (ASR) resources. For information about configuring the Media Resource Function, see *Overview of configuring the Media Resource Function* on page 39.

The following illustration shows the relationships among the Vision VoiceXML Interpreter, SSML Processor, and Media Resource Function in the VoiceXML Subsystem:



VoiceXML Subystem

Audio and video codecs

The VoiceXML Subsystem supports the following audio and video codecs:

Codec type	Description
Audio	• AMR (3GPP TS 26.090, 26.101, and 26.073, version 5.3.0, 2004) • G.711 A-law and mu-law • G.723.1 • G.726 • G.729 A
Video	• H.263, H.263+ • H.264 • MPEG-4

Real time protocol (RTSP) streaming servers

The VoiceXML Subsystem supports the following RTSP streaming servers:

Vendor	Product
Real	Helix Mobile Server 11.0
Apple	Darwin Streaming Server 5.5

Installation

Each component of the VoiceXML Subsystem is a Linux daemon process installed as a service. The components start automatically when the VX Integrated Media Platform starts. If you need to troubleshoot issues with the VX Integrated Media Platform, you can either use the Vision™ Console to start and stop each component, or you can start and stop each component manually. For information, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

Documentation conventions

This manual has the following conventions:

- The VX Integrated Media Platform software is installed in the */opt/nms/vx/* directory. This manual uses the string *vx* to refer to this installation directory.
- In the settings tables, the term *default value* refers to the value used when no value is specified. The term "initial value" refers to the value set by the manufacturer, when this value differs from the default value. Unless otherwise specified, the initial value is the same as the default value.

System file locations

The main VoiceXML Subsystem system files are stored in the following default locations:

Directory	Description
vx/vxmlinterpreter/conf/vxmlinterpreter.conf	VoiceXML Interpreter configuration file, which configures settings for the VoiceXML Interpreter and the SIP capabilities of the Media Resource Function. For more information, see *Overview of configuring the VoiceXML Interpreter* on page 11 and *Configuring MIOSIP* on page 40.
vx/vxmlinterpreter/conf/mrcp.xml	VoiceXML Interpreter MRCP configuration file, which configures automatic speech recognition (ASR) engines. For more information, see *Configuring ASR resources* on page 52.
vx/ssmlprocessor/conf/ssmlprocessor.conf	SSML Processor configuration file, which configures settings for the SSML Processor and the audio capabilities of the Media Resource Function. For more information, see *Overview of configuring the SSML Processor* on page 31 and *Configuring ADTCP* on page 50.
vx/vxmlinterpreter/conf/voicexmlappcfg.xml	Application configuration file, which the VoiceXML Interpreter process reads at startup to populate the list of VoiceXML applications. For more information, see *Application configuration file* on page 29.
vx/ssmlprocessor/conf/mrcp.xml	SSML Processor MRCP configuration file, which configures MRCP text-to-speech (TTS) engines. For more information, see *Configuring MRCP TTS resources* on page 54.

Directory	Description
vx/ssmlprocessor/conf/config.xml	Engine configuration file, which does all of the following: • Allocates the media engine used for audio and video playback. • Reserves the resources needed to use MRCP TTS engines. • Reserves the resources needed to use non-MRCP TTS engines. • Specifies required settings for non-MRCP TTS engines. For more information, see *Configuring MRCP TTS resources* on page 54 and *Configuring non-MRCP TTS resources* on page 55.

References to standards

The following table provides references to the standards used by the VoiceXML Subsystem:

Standard	Version
MRCP	*A Media Resource Control Protocol Developed by Cisco, Nuance, and Speechworks*, Shanmugham, Monaco, and Eberman, IETF Internet-Draft, draft-shanmugham-mrcp-05, January 2004.
RFC 2833	*RTP Payload for DTMF Digits, Telephony Tones and Telephony Signal*, Schulzrinne and Petrack, IETF RFC 2833, May 2000.
RTSP	*Real Time Streaming Protocol (RTSP), Schulzrinne, Rao, and Lanphier*, IETF RFC 2326, April 1998.
SIP	*SIP: Session Initiation Protocol*, Rosenberg *et al*., IETF RFC 3261, June 2002
SIP-VXML	*SIP Interface to VoiceXML Media Services*, Burke *et al*., IETF Internet-Draft, draft-burke-vxml-02, November 2006.
SRGS	*Speech Recognition Grammar Specification Version 1.0*, Hunt et al, W3C Candidate Recommendation, 16th March 2004. http://www.w3.org/TR/speech-grammar/
SSML	*Speech Synthesis Markup Language Version 1.0*, Burnett et al, W3C Recommendation, 7th September 2004. http://www.w3.org/TR/speech-synthesis/
VoiceXML	*Voice Extensible Markup Language (VoiceXML) Version 2.0*, McGlashan et al, W3C Recommendation, 16th March 2004. http://www.w3.org/TR/voicexml20/
	Voice Extensible Markup Language (VoiceXML) Version 2.1, Oshry et al, W3C Recommendation, 13th June 2005. http://www.w3.org/TR/voicexml21/

3 Configuring the VoiceXML Interpreter

Overview of configuring the VoiceXML Interpreter

This topic describes how to configure the VoiceXML Interpreter by editing the VoiceXML Interpreter configuration file. The VoiceXML Interpreter configuration file is named *vxmlinterpreter.conf*, and it resides in the vx/*vxmlinterpreter/conf* directory.

When specifying values for a particular setting, use the following syntax:

```
setting-name=setting-value
```

If ***setting-value*** is a long string, spread the value over several lines by placing a backslash (\) at the end of every line.

To designate time, use either seconds (s) or milliseconds (ms) unless otherwise stated.

You can specify the following types of VoiceXML Interpreter settings:

- General settings
- Logging settings
- Telephony settings
- Proxy cache settings
- Local cache settings
- CallPlacer settings
- Billing server settings
- Locale-specific settings
- Runtime settings

Note: For information about configuring the VoiceXML Interpreter using the Vision™ Console, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

VoiceXML Interpreter general settings

Use the following settings to specify general settings for the VoiceXML Interpreter:

Setting	Description
AllowCallTransfer	Specifies whether call transfers are allowed. This behavior can be overridden at runtime using the property com.vision.transferallowed. Valid values: • true • false Default: true

Setting	Description
ConnectionProtocolName	Sets the value for the VoiceXML session variable session.connection.protocol.name if no value for this is passed up by the MIO. Default: None
ConnectionProtocolVersion	Sets the value for the VoiceXML session variable session.connection.protocol.version if no value for this is passed up by the MIO. Default: None
DefaultAudioMaxAge	Sets the default value of the VoiceXML audiomaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None
DefaultAudioMaxStale	Sets the default value of the VoiceXML audiomaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0
DefaultBargeinType	Sets the default value of the VoiceXML bargeintype property. Valid values: • speech • hotword Default: speech
DefaultCompleteTimeout	Sets the default value of the VoiceXML completetimeout property. Valid values: 0.2s - 10s Default: 0.25s
DefaultDataMaxAge	Sets the default value of the VoiceXML datamaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None
DefaultDataMaxStale	Sets the default value of the VoiceXML 2.1 datamaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0
DefaultDocumentMaxAge	Sets the default value of the VoiceXML documentmaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None

Setting	Description
DefaultDocumentMaxStale	Sets the default value of the VoiceXML documentmaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0
DefaultFetchAudioDelay	Sets the default value of the VoiceXML fetchaudiodelay property Valid values: 0s - ***n***s Default: 0s
DefaultFetchAudioMinimum	Sets the default value of the VoiceXML fetchaudiominimum property. Valid values: 0s - ***n***s Default: 0s
DefaultFetchTimeout	Sets the default value of the VoiceXML fetchtimeout property. Valid values: 0s - ***n***s Default: 10s
DefaultGrammarMaxAge	Sets the default value of the VoiceXML grammarmaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None
DefaultGrammarLocale	Sets the default RFC 3066 language identifier to use for grammars. Default: None Initial value: en-GB
DefaultGrammarMaxStale	Sets the default value of the VoiceXML grammarmaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0
DefaultIncompleteTimeout	Sets the default value of the VoiceXML incompletetimeout property. Valid values: 0.2s - 10s Default: 0.75s
DefaultInitialURI	URI of the initial page to execute when: • An incoming call is answered, if the dialed number does not match the range of any configured VoiceXML application. • Placing a call, if the service number does not match the number range of any configured VoiceXML application. The value must be a full URI, because relative URIs are not allowed. Both HTTP and local file URIs are supported. In the latter case, the *file://* protocol specifier must precede the path. Default: None
DefaultInterDigitTimeout	Sets the default value of the VoiceXML interdigittimeout property. Valid values: 0 - 600s Default: 3s

Setting	Description
DefaultMaxSpeechTimeout	Sets the default value of the VoiceXML maxspeechtimeout property. Valid values: 0.05s - 600s Default: 15s
DefaultObjectMaxAge	Sets the default value of the VoiceXML objectmaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None
DefaultObjectMaxStale	Sets the default value of the VoiceXML objectmaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0
DefaultRecordMaxtime	Sets the default value of the maxtime attribute for recordings. Valid values: 0s - 600s Default: 60s
DefaultRecordType	Sets the default value for the VoiceXML 2.1 recordutterancetype property and the type attribute of the <record> element. Default: audio/x-wav
DefaultRecordUtterance	Specifies the default value for the VoiceXML recordutterance property. Valid values: • true • false Default: false For more information about using record utterance functionality in a VoiceXML application, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.
DefaultScriptMaxAge	Sets the default value of the VoiceXML scriptmaxage property. The value must be an integer, as per the VoiceXML specification. When a value is not specified, the parameter is ignored when fetching the resource. Valid values: 0 - ***n*** Default: None
DefaultScriptMaxStale	Sets the default value of the VoiceXML scriptmaxstale property. The value must be an integer, as per the VoiceXML specification. Valid values: 0 - ***n*** Default: 0

Setting	Description
DefaultStrictVXML	Specifies the default value of the com.vision.strictvxml runtime setting. Valid values: • true • false Default: false
DefaultTimeout	Sets the default value of the VoiceXML timeout property. Valid values: 0.05s - 20000s Default: 3.4s
DefaultTTSLang	Sets the default value for the com.vision.ttslang property. The value should be a language-identifier as per RFC 3066. It can have a particular voice name appended, for example, en-GB-Crystal. Default: None Initial value: en-GB
ExecutionCounterCheckMultiple	A multiple at which the VoiceXML Interpreter forces prompts to play. Using this setting can avoid performance decreases when the VoiceXML Interpreter encounters an application with a tight loop, because it gives the user an opportunity to hang up. Valid values: 0 - 1000 Default: 100
ExecutionCounterMax	Value of the execution counter at which the VoiceXML Interpreter should end execution. A value of 0 indicates no limit on the execution counter. With this value, a call never terminates due to the execution counter. Valid values: 0 - (2^{32} -1) Default: 500
ExitDataXML	Indicates whether data used in the VoiceXML <exit> expr and namelist attributes is in XML form or is encoded per SIP-VXML. If true, this setting enables backwards-compatibility with systems that require this data in XML form. Valid values: • true: Data stays in XML form. • false: Data is encoded. Default: false For more information about SIP-VXML, see *References to standards* on page 10.

Setting	Description
ForceVXMLValidation	Forces the VoiceXML Interpreter to validate all VoiceXML documents using the relevant DTD. By default, the VoiceXML Interpreter performs the initial document parse without validating. If the interpreter subsequently discovers an error, it re-parses the document with validation enabled. Setting this value to true causes the interpreter to validate documents on the initial parse. Validation provides robust error checking on VoiceXML document structure and syntax, and gives detailed descriptions of any errors found. You should use the default (false) in a production environment, because validating documents on the initial parse can decrease performance under high loads. Valid values: • true • false Default: false
InitialDocumentMaxAge	Value of the maxage directive to use when fetching the initial VoiceXML document. The default value is 0, which ensures that the interpreter fetches a fresh copy at the start of each call. Valid values: 0 - ***n*** Default: 0
LicenseDir	Location of the license files for the VoiceXML Subsystem. The VX Integrated Media Platform searches this directory for any files ending with the extension *.lic*, and then attempts to validate each file as a license. Default: *vx/license*
MaxConcurrentOutboundCalls	Maximum number of concurrent outbound calls allowed. VoiceXML interpreter channels are no longer configured to allow only inbound or only outbound calls. A single channel can be selected for either type of call. When an outbound call is requested, a reverse search of the channels is performed until a free channel is found. When an incoming call is received, an available channel is found using a forward search. This minimizes contention. Valid values: 1 - 512 Default: 512 - No limit on the number of concurrent outbound calls. All channels can be used for outbound calls at any time.
MaximumThrowRecursionLevel	Maximum number of recursive events that can be generated before execution is aborted. Valid values: 5 - 300 Default: 75
MIOImplName	Name of the Media Interface Object (MIO) implementation that the VoiceXML Interpreter uses to perform prompt playback, speech recognition, call transfers, and so on. This is the name of the MIO DLL that the VoiceXML Interpreter loads at startup, excluding the file extension. For example, to use *miosip.dll* or *miosip.so*, set this parameter to miosip. Default: None Initial value: miosip

Setting	Description
MIOImplParams	Space-separated list of the vendor-specific parameters used to configure the Media Interface Object (MIO). These parameters are set at initialization time and cannot be set at runtime using VoiceXML properties. Default: None For more information, see *Configuring MIOSIP* on page 40.
NativeSessionOnly	Configures the VoiceXML Interpreter to run as a native application using the local machine's sound card as the input/output device (as opposed to a telephony card or Voice over IP). This configuration is mainly used for development. In this configuration, only one instance is started (the NumChannels setting is ignored) and document execution begins immediately after the VoiceXML Interpreter starts up. Valid values: • true • false Default: false
NumChannels	Number of VoiceXML Interpreter channels to be started. Each channel runs as a separate thread within the VoiceXML Interpreter executable. Valid values: 1 - 240 Default: Based on the number of licensed ports.
ProcessName	Name of the VoiceXML Interpreter process. Default: vxmlinterpreter
RecordPrompts	Specifies the default value for the property com.vision.recordprompts. Valid values: • true: Prompt files used during the call are copied from the VoiceXML Interpreter's cache to the call log directory at the end of the call. • false: Prompt files are not copied at the end of the call. Default: false
SNMPEnabled	Specifies whether to enable the VoiceXML Interpreter's SNMP sub-agent. The SNMP sub-agent handles SNMP requests for getting and setting VoiceXML Interpreter management information. It also sends SNMP notifications through the Emanate SNMP master-agent. Valid values: • true • false Default: true For more information about using SNMP with the VX Integrated Media Platform, see the *Dialogic® Vision™ SNMP Reference Manual*.
StaticContentDir	Location of static content for the VoiceXML Interpreter's web server. Default: *vx/vxmlinterpreter/www*

Setting	Description
SupportedLocales	Comma-separated list of RFC 3066 language identifiers specifying the locales supported by the current configuration. For each locale, a configuration file can be specified containing locale-specific configuration settings. For example, ApproximateGrammarOmitWords. The configuration file is named *vxmlinterpreter_**xx-XX**.conf*, where ***xx-XX*** is the language identifier. The file is located in the ***xx-XX*** directory. For example, *vx/vxmlinterpreter/data/en-GB/vxmlinterpreter_en-GB.conf*. When the VoiceXML Interpreter starts, it attempts to load the configuration file for each of the locales specified in this setting. Default: None
SystemMaxCallDurationLimit	Maximum allowed length of a call. Setting it to 0 indicates there is no limit on the length of a call. In this case, other settings, such as com.vision.maxcallduration, can override the value of 0. Valid values: 0s - ***n***s Default: 1200s
VXMLAppCfg	URI of the VoiceXML application configuration file. This document contains mappings of number ranges to applications. It also contains settings for the application, such as whether call transfers are allowed and maximum call duration. For more information, see *VoiceXML Interpreter application management* on page 28. Only file:// URIs are supported. Default: *file:///vx/vxmlinterpreter/conf/voicexmlappcfg.xml*
VXMLObjects	Comma-separated list of VoiceXML <object> implementations to be loaded. Set each value to the name of a shared object located in the same directory as the executable. Exclude the file extension when setting this value. For example, to use the default VXMLObject implementation (*vxmlobject.dll*), include vxmlobject in this parameter. Default: None Initial value: VODTMFGenerator

VoiceXML Interpreter logging settings

Use the following settings to configure logging for the VoiceXML Interpreter:

Setting	Description
LogDir	Directory to which VoiceXML Interpreter system logs are written. Default: *vx/vxmlinterpreter/logs* For information on VoiceXML Interpreter logging, see the *Vision Dialogic Vision™ VX Integrated Media Platform User's Manual*.
SystemLogFileMaxNum	Maximum number of files allowed in the VoiceXML Interpreter system log directory at any time. When this limit is reached, the VoiceXML Interpreter process deletes older files as new files are created. Valid values: 1 - 9999 Default: 50

Setting	Description
SystemLogFileMaxSize	Maximum size of the VoiceXML Interpreter system log file before a new log file is started with the file index incremented. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Valid values: 100KB - 500MB Default: 10MB
SystemLogLevel	Minimum severity level that must be assigned to a VoiceXML application log message for it to be written to the VoiceXML Interpreter system log file. If the value specified is less than 4, VoiceXML application log messages are not written to the system log file. Valid values: • 1-5, where 5 represents the highest verbosity • WARNING • ERROR • FATAL Default: 1 **Note:** You can also change the log level by using the Vision™ Console. For more information, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.
SystemLogTime	Time format for the VoiceXML Interpreter system log and the VoiceXML application log. Valid values: • GMT: Greenwich Mean Time format. • LOCAL: Same time format as the local VX Integrated Media Platform. Default: LOCAL
VXMLAppLogBaseDir	Base directory to which VoiceXML application logs are written. If specified, the com.vision.voicexmllog property provides the remainder of the path and is appended to the base directory. If VXMLAppLogBaseDir is not set, the value defaults to the value of the LogDir setting. Default: None Initial value: *vx/vxmlinterpreter/logs*
VXMLAppLogMaxAge	Number of hours for which the VoiceXML Interpreter keeps application logs, expressed as an integer. Leave the default value (0) if you do not want the VoiceXML Interpreter to prune application logs based on age. Default: 0
VXMLAppLogMaxSize	Maximum amount of disk space used by the application logs. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Set this value to 0 if you do not want the VoiceXML Interpreter to prune application logs based on disk space usage. Default: 250MB.

Setting	Description
VXMLAppLogsEnabled	Specifies whether to enable VoiceXML application logging. Valid values: • true: Logs are created for a particular application based on that application's configuration. • false: Application logs are not created. Default: true For more information, see *VoiceXML Interpreter application management* on page 28.

VoiceXML Interpreter telephony settings

Use the following settings to configure the telephony environment that the VoiceXML Interpreter uses:

Setting	Description
DefaultCallConnectTimeout	Time period after which a NOANSWER is returned when attempting to connect on an outbound call or transfer. A value of 0 indicates that no timeout occurred. Override this by setting the timeout explicitly in the telephone number as shown in this example: `tel:+353-1-2345678;timeout=15s` Valid values: 0s - ***n***s Default: 30s In the case of a call transfer, the developer can override the DefaultCallConnectTimeout by using the VoiceXML connecttimeout attribute.
DefaultOutboundAni	ANI to use for outbound calls. Valid values: • ServiceNumber • Any arbitrary number Default: ServiceNumber The developer can override this by setting the ANI explicitly in the telephone number as shown in this example: `35312345678;ani=5557755`
DefaultTransferAni	ANI to use for transfers. Valid values: • ServiceNumber • OriginalANI • Any arbitrary number Default: ServiceNumber The developer can override this by setting the ANI explicitly in the telephone number as shown in this example: `35312345678;ani=5557755`

VoiceXML Interpreter proxy cache settings

Use the following settings to enable the VoiceXML Interpreter to use a proxy cache when requesting resources over HTTP:

Setting	Description
BypassProxyAddresses	Comma-separated list of internet addresses for which the proxy is to be bypassed. Default: 127.0.0.1, localhost
ProxyAddress	Name or IP address of the VoiceXML Interpreter proxy server. If not set, no proxy is used. Default: None
ProxyPort	VoiceXML Interpreter port of the proxy server. Default: 3128

VoiceXML Interpreter local cache settings

The VX Integrated Media Platform uses a memory cache and a disk cache for the following types of resources:

Resource type	What the related memory and disk caches store
Grammar	Grammar files used by VoiceXML documents. These files can be inline with the VoiceXML or requested and downloaded from an HTTP web server.
Script	Script files used by VoiceXML documents. These files can be inline with the VoiceXML or requested and downloaded from an HTTP web server.
VoiceXML	VoiceXML files requested and downloaded from either of the following: • An HTTP web server • A local file URI, using the cache settings specified by the web server
XML	XML data fetched using the VoiceXML 2.1 <data> element.

If a cached resource is read from the disk back into the memory cache, the file remains on the disk. Therefore, an item in the memory cache might use up disk space as well as memory. The default local cache settings take this into account.

Use the following settings to configure the resource manager and the local cache for the VoiceXML Interpreter.

Setting	Description
CacheDir	Directory to which cached resources are written when serializing them to disk. Default: *vx/vxmlinterpreter/cache*
GrammarCacheMaxSizeDisk	Maximum allowed size of the disk cache for grammar files. When the cache size exceeds this value, the least recently used grammar files are deleted from the disk. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. A value of 0 indicates that no grammar resources are written to disk. Default: 0MB
GrammarCacheMaxSizeMemory	Maximum allowed size of the memory cache for grammar files. When the cache size exceeds this value, the least recently used grammar resources are deleted from memory. These files are serialized to disk if the value of GrammarCacheMaxSizeDisk is greater than zero. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 10MB
ResourceManagerHeuristicFraction	Value of the heuristic fraction used when calculating a value for the max-age of cached resources, where one of the following HTTP header values is set: • no expires • cache-control: max-age • cache-control: s-maxage The algorithm used is described in the HTTP 1.1 specification (RFC 2616). Valid values: 0.0 - 1.0 Default: 0.1
ResourceManagerHeuristicMaximum	Value of the heuristic maximum used in the ResourceManagerHeuristicFraction setting. Valid values: 0 - ***n*** Default: 86400
ScriptCacheMaxSizeDisk	Maximum allowed size of the disk cache for script files. When the cache size exceeds this value, the least recently used script files are deleted from the disk. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. A value of 0 indicates that no script files are written to disk. Default: 0MB

Setting	Description
ScriptCacheMaxSizeMemory	Maximum allowed size of the memory cache for script files. When the cache size exceeds this value, the least recently used script files are deleted from memory. These files are serialized to disk if the value of ScriptCacheMaxSizeDisk is greater than zero. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 10MB
XMLCacheMaxSizeDisk	Maximum allowed size of the disk cache for XML resources. When the cache size exceeds this value, the least recently used XML resources are deleted from the disk. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. A value of 0 indicates that no XML files are written to disk. Default: 0MB
XMLCacheMaxSizeMemory	Maximum allowed size of the memory cache for XML resources fetched using the VoiceXML 2.1 <data> element. When the cache size exceeds this value, the least recently used XML files are deleted from memory. These files are serialized to disk if the value of XMLCacheMaxSizeDisk is greater than zero. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 10MB
VXMLCacheMaxSizeDisk	Maximum allowed size of the disk cache for VoiceXML files. When the cache size exceeds this value, the least recently used VoiceXML files are deleted from the disk. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 0MB
VXMLCacheMaxSizeMemory	Maximum allowed size of the memory cache for VoiceXML resources. When the cache size exceeds this value, the least recently used VoiceXML files are deleted from memory. These files are serialized to disk if the value of VXMLCacheMaxSizeDisk is greater than zero. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 10MB

VoiceXML Interpreter CallPlacer settings

The CallPlacer processes outbound call requests received through the web server. To configure the CallPlacer component of the VoiceXML Interpreter, use the following setting:

Setting	Description
CallPlacerStaggerDcValue	Average number of milliseconds to wait between outbound call initiations. Valid values: 0 - 10000 Default: 500

For more information about the CallPlacer, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

VoiceXML Interpreter billing server settings

To configure the VoiceXML Interpreter to run in conjunction with a billing server, use the following settings. The settings apply only if you set BillingEnabled to true.

Setting	Description
BillingAuthoriseOnOSPUriFail	Whether or not to authorize calls when all the URIs specified in BillingOSPURIs are not available. Valid values: • true • false Default: true
BillingAuthoriseUnknownInbound	Whether or not to authorize inbound calls when the DNIS is not known by the billing server. Valid values: • true • false Default: false
BillingAuthoriseUnknownOutbound	Whether or not to authorize outbound calls when the DNIS is not known by the billing server. Valid values: • true • false
BillingCDRBaseDir	Directory in which to store call detail records (CDRs). CDRs are written every hour to a file named ***YYYY*/*MM*/*DD*/*HH***.*cdr*, off the base directory. CDRs written to this file are in a concise, non-XML format. Default: *vx/vxmlinterpreter/cdr*

Setting	Description
BillingConnectionTimeout	Sets a timeout in seconds for authorizing a call and sending a call detail record (CDR) for a call. If a connection with the billing server is not established in this number of seconds, the request times out and the connection fails. A value of 0 indicates that the request never times out. If a connection to the billing server succeeds, keep-alive is performed on the connection on a per-channel basis. **Note:** If the BillingConnectionCDRTimeout setting is set, then BillingConnectionTimeout applies only to authorization connections. Valid values: 0 - 100 Default: 10
BillingConnectionCDRTimeout	Enables you to set different timeouts for authorizing a call and sending a call detail record (CDR) for a call. If used, this setting determines the time period after which an attempt to send a CDR to the billing server times out and the connection fails. A value of 0 indicates that the request never times out. If not used, the value of BillingConnectionTimeout setting determines the timeout value for sending CDRs. Valid values: 0 -100 Default: 10
BillingEnabled	Whether the VoiceXML Interpreter is used in conjunction with a billing server. Valid values: • true • false Default: false
BillingOmitUsageDetail	Whether <UsageDetail> elements are omitted from usage indications sent to the OSP billing server. Valid values: • true • false Default: false
BillingOSPURIs	A comma-separated list of the URIs of OSP billing servers to use. If more than one URI is specified and the first is unavailable, the VoiceXML Interpreter tries each of the URIs in turn, until a successful connection is established or the list is exhausted. Default: None

Setting	Description
BillingUseMasqueradeCache	Whether the billing client does the following: • Looks up the service number corresponding to the current DNIS in its local cache. • Uses that value, if present, rather than attempting to connect to the billing database each time. Valid values: • true • false Default: true
BillingWriteCDRToDisk	Determines when CDRs are written to disk. Valid values: • true: CDRs are written to disk regardless of whether they were successfully uploaded to the OSP billing server. • false: CDRs are written to disk only if the upload to the OSP server fails or if no OSP URIs are specified in the configuration file. Default: false **Note:** CDRs that were not successfully uploaded are preceded by the @ character when written to the file. This allows a third party CDR upload utility to identify these CDRs when scanning the file.

VoiceXML Interpreter locale-specific settings

Use the ApproximateGrammarOmitWords setting to specify non-critical words for a particular locale. The setting must be included in the configuration file for the locale. For example, to specify the words that are non-critical for the English locale, en-GB, specify the parameters in the file, *vx/vxmlinterpreter/data/en-GB/vxmlinterpreter_en-GB.conf*.

Setting	Description
ApproximateGrammarOmitWords	Comma-separated list of words to treat as non-critical when generating an approximate grammar for a <choice> or <option>. The resultant grammar does not require that these words be spoken. For example, for an English locale such as en-GB, the file *vxmlinterpreter_en-GB.conf* might contain the following setting: `ApproximateGrammarOmitWords=the, a, and` If no words are specified, all words are treated equally. Default: None

VoiceXML Interpreter runtime settings

In addition to the standard properties described in the VoiceXML specification, the VoiceXML Interpreter supports platform-specific properties. Many of the platform-specific properties have default values that can be set in the configuration file. Some runtime properties can override these initialization settings. The following table describes the properties that can be set at runtime:

Setting	Description
com.vision.asrengine	Specifies the VoiceXML ASR engine to use. The Media Interface Object (MIO) determines the default. For more information, see *Configuring MIOSIP* on page 40 and *Configuring ASR resources* on page 52. Default: None
com.vision.maxcallduration	Maximum duration of a call. If the call has already exceeded the specified time when the property is set, it is terminated immediately. Valid values: 0s - ***n***s Default: Specified by the VoiceXML application. However, the SystemMaxCallDurationLimit setting (if it is not set to 0) can override the default. For information about the SystemMaxCallDurationLimit setting, see *VoiceXML Interpreter general settings* on page 11.
com.vision.maxcalldurationlimit	Upper limit for the com.vision.maxcallduration setting. You can set this property only once per call. Valid values: 0s - ***n***s Default: Specified by the SystemMaxCallDurationLimit setting. For information about the SystemMaxCallDurationLimit setting, see *VoiceXML Interpreter general settings* on page 11.
com.vision.maxcalldurationprompt	URL of a prompt to play when the maximum call duration is reached. The VoiceXML Interpreter plays the prompt, then terminates the call. Default: None (plays *builtin:audio/calltimeout)*.
com.vision.strictvxml	Specifies whether strict VoiceXML syntax checking is performed. Valid values: • true: An error.badfetch is generated if a VoiceXML document does not specify a version or xmlns attribute, or if the value of the xmlns attribute is not *http://www.w3.org/2001/vxml.* • false: Strict VoiceXML checking is not performed. Default: false **Note:** Setting com.vision.strictvxml to true can slow system performance.

Setting	Description
com.vision.voicexmllog	Only applies if the application being executed is not one of the VoiceXML Interpreter's managed applications. Specifies the subdirectory to which the application (VoiceXML) log file for the current call is written. The value is appended to that of the VXMLAppLogBaseDir setting to obtain the complete path. This property can only be set once per call. If not set, no application logs are written for the call. Default: None For information about the VXMLAppLogBaseDir setting, see *VoiceXML Interpreter logging settings* on page 18.

VoiceXML Interpreter application management

The VoiceXML Interpreter manages VoiceXML applications. When an incoming call is received, the caller is automatically redirected to the initial URI of a particular application based on the dialed number (DNIS). For outbound calls, the application is determined by the service number, if any, specified in the CallPlacer XML request. If a call is received whose DNIS does not match any of the configured applications, the caller is redirected to the URI specified in the DefaultInitialURI setting in the *vxmlinterpreter.conf* file.

Manage applications by using the following tools:

- Vision™ Console
- SNMP
- Application configuration file

Vision™ Console

The Vision™ Console is a web-based tool that lets you specify the following properties for individual VoiceXML applications:

- Number range that maps to the application
- Initial URI to use for an incoming call, based on the specified number range
- Whether logging is enabled, and the name of the application log file
- Whether call transfers are allowed
- Maximum call duration

For information about using the Vision™ Console, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

SNMP

Use SNMP with the VX Integrated Media Platform to get information about each managed VoiceXML application including configuration and statistical information. The configuration information includes the application’s number range, initial URI, and maximum call duration. The statistical information includes the total number of calls to the application, average call duration, recognition success/failure rates, and number of errors generated.

For information about using SNMP, see the *Dialogic® Vision™ SNMP Reference Manual.*

Application configuration file

The application configuration file is an XML file that the VoiceXML Interpreter reads at startup to populate the list of applications. The location of the application configuration file is specified by the VXMLAppCfg setting in the *vxmlinterpreter.conf* file, and defaults to *file:///opt/nmsvx/vxmlinterpreter/conf/voicexmlappcfg.xml.*

Note: Only file URIs are supported for this setting. HTTP is not supported.

To ensure that changes persist across restarts of the process, the XML application configuration file is re-written anytime an application is added, modified, or removed. Consequently, do not modify this file while the VoiceXML Interpreter process is running, as any subsequent changes made using the Vision™ Console or SNMP cause the changes to be overwritten.

Sample configuration file

The following example shows an application configuration file:

```
<?xml version="1.0"?>
<voicexml-app-config version="1.0">
<application number-range=".%">
    <initialuri> http://localhost:9002/vxml/examples/index.vxml</initialuri>
    <logging enabled="true">
        <subdir></subdir>
    </logging>
    <transfer allowed="false"/>
    <maxcallduration>300s</maxcallduration>
</application>
</voicexml-app-config>
```

The following table describes the contents of the application configuration file:

Element	Attribute	Description
<application>	number range	VoiceXML application you are defining. The document root element contains an <application> child element for each VoiceXML application. The number range attribute specifies the range of numbers that map to the application.
<initialuri>	init	URI of the initial document that the VoiceXML Interpreter loads when the call is answered.
<logging>	enabled	A Boolean expression indicating whether or not to enable logging for this application. Applies only if VXMLAppLogsEnabled is set to true in the VoiceXML Interpreter configuration file (*vxmlinterpreter.conf*).
<subdir>		Subdirectory to which log files for this application are written. The value is a relative file path (empty by default) that is appended to the value specified by the VXMLAppLogBaseDir setting in the VoiceXML Interpreter configuration file (*vxmlinterpreter.conf*). Obtain the complete path by appending directories corresponding to the current year, month, day, and hour. For more information, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.
<transfer>	allowed	A Boolean expression indicating whether call transfers are allowed for this application.
<maxcallduration>		A time designator indicating the maximum allowed duration of any call made to this application. The value must include an s (for seconds) or an ms (for milliseconds).

4 Configuring the SSML Processor

Overview of configuring the SSML Processor

This topic descries how to configure the SSML Processor by editing the following configuration files:

- SSML Processor configuration file, which specifies the initialization time settings for the SSML Processor. This file is named *ssmlprocessor.conf*.
- Engine configuration file, which specifies the location and settings of the audio and text-to-speech (TTS) engines used by the SSML. This file is named *config.xml*.
- SSML Processor MRCP configuration file (optional), which specifies information for the available MRCP TTS engines. This file is named *mrcp.xml*.

These configuration files reside in the *vx/ssmlprocessor/conf* directory.

When specifying values for a particular setting, use the following syntax:

```
setting-name=setting-value
```

If ***setting-value*** is a long string, spread the value over several lines by placing a backslash (\) at the end of every line.

To designate time, use either seconds (s) or milliseconds (ms) unless otherwise stated.

You can specify the following types of SSML Processor settings:

- General settings
- Logging settings
- Proxy cache settings
- Local cache settings
- Media engine settings
- MRCP settings

Note: For information about configuring the SSML Processor using the Vision™ Console, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

SSML Processor general settings

Use the following settings in the *ssmlprocessor.conf* file to specify general settings for the SSML Processor:

Setting	Description
ADImplName	Name of the Audio Driver (AD) implementation that the SSML Processor uses to communicate with the underlying media platform. This is the name of the AD DLL or shared object that the Processor loads at start-up (excluding the file extension). For example, to use *adtcp.dll* or *adtcp.so*, set this parameter to adtcp. Default: None. Initial value: adtcp
ADImplParams	Space-separated list of the vendor-specific parameters used to configure the audio driver. Default: None Initial value: com.vision.TTSListenPort=32323 For more information, see *Configuring ADTCP* on page 50.
DefaultAudioCodec	Codec the SSML Processor uses when initializing TTS engine instances. Individual SSML requests can override this setting. In this situation, the SSML Processor performs a mu-law to A-law or A-law to mu-law conversion on the data, before sending it to the client. Audio data is always normalized to 8 kHz, 8-bit PCM format. Valid values: • alaw • mulaw Default: alaw
DefaultLocale	Default locale to use when rendering speech for which no xml:lang is specified. Default: None Initial value: en-GB
EngineConfig	Location of the engine configuration file, an XML file that specifies the location and settings of the audio and TTS engines used by the SSML Processor. Default: *vx/ssmlprocessor/conf/config.xml* For more information, see *Configuring MRCP TTS resources* on page 54, and *Configuring non-MRCP TTS resources* on page 55.
EngineFixerTriggerInterval	Interval by which the SSML Processor retries a connection to a previously failed non-MRCP TTS engine. Default: 1800 For more information, see *Configuring non-MRCP TTS resources* on page 55.
ProcessName	Name of the SSML Processor process. Default: ssmlprocessor

Setting	Description
SSMLErrorLevel	Circumstances under which the SSML Processor notifies the VoiceXML Interpreter of an error during document execution. This type of notification occurs through an error.noresource event. Valid values: • none: SSML Processor never notifies the VoiceXML Interpreter about an error. • tts: SSML Processor notifies the VoiceXML Interpreter if it fails to find or connect to a TTS engine. • media: SSML Processor notifies the VoiceXML Interpreter if a <media> or <audio> element fails to render, and there is no alternate content. • all: SSML Processor notifies the VoiceXML Interpreter if it fails to find or connect to a TTS engine, or if a <media> or <audio> element fails to render, and there is no alternate content. Default: tts
StaticContentDir	Location of static content for the SSML Processor's web server. Default: None Initial value: *vx/ssmlprocessor/www*
UserAgent	User-Agent request header when performing HTTP fetches. Default: vision-browser/3.5

SSML Processor logging settings

Use the following settings in the *ssmlprocessor.conf* file to configure logging for the SSML Processor:

Setting	Description
LogDir	Directory to which system log files are written. Default: *vx/ssmlprocessor/logs* For information on SSML Processor logging, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.
SystemLogLevel	Minimum severity level that must be assigned to a log message for it to be written to the system log file. Valid values: • 1 - 5, where 5 represents the highest verbosity • WARNING • ERROR • FATAL Default: 1 **Note:** You can also change the log level by using the Vision™ Console. For more information, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

Setting	Description
SystemLogFileMaxNum	Maximum number of files allowed in the log directory at any time. When this limit is reached, older files are deleted by the SSML Processor as new files are created. Valid values: 1 - 9999 Default: 50
SystemLogFileMaxSize	Maximum size of the system log file before a new log file is started with the file index incremented. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Valid values: 100KB - 500MB Default: 10MB
SystemLogTime	Time format for the SSML Processor system log. Valid values: • GMT: Greenwich Mean Time format. • LOCAL: Same time format as the local VX Integrated Media Platform. Default: LOCAL

SSML Processor proxy cache settings

Use the following settings in the *ssmlprocessor.conf* file to enable the SSML Processor to use a proxy cache when requesting resources over HTTP:

Setting	Description
BypassProxyAddresses	Comma-separated list of internet addresses for which the proxy is to be bypassed. Default: 127.0.0.1, localhost
ProxyAddress	Name or IP address of the SSML Processor proxy server. If not set, no proxy is used. Default: None
ProxyPort	SSML Processor port of the proxy server. Default: 3128

SSML Processor local cache settings

The SSML Processor uses a memory cache and a disk cache for the following types of resources:

For this resource type...	The related memory and disk caches store...
Lexicon	Lexicon resources used by an SSML request. Lexicons contain pronunciation information for particular words in the SSML request.
Prompt	Audio/multimedia files requested and downloaded from an HTTP web server, using cache settings specified by the web server.

If a cached resource is read from the disk back into the memory cache, the file remains on the disk. Therefore, an item in the memory cache might use up disk space as well as memory. The default local cache settings take this into account.

Use the following settings in the *ssmlprocessor.conf* file to configure the resource manager and the local cache for the SSML Processor:

Setting	Description
CacheDir	Directory to which to write cached files when serializing them to the disk cache. Default: v*x/ssmlprocessor/cache*
LexiconCacheMaxSizeDisk	Maximum allowed size of the disk cache for lexicon files. When the maximum disk size exceeds this value, the least recently used lexicon resources are removed from the disk. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. A value of 0 indicates that no lexicon files are written to disk. Default: 25MB Initial value: 50MB
LexiconCacheMaxSizeMemory	Maximum allowed size of the memory cache for the lexicon files. When the cache size exceeds this value, the least recently used lexicon resources are deleted from memory. These files are serialized to disk if the value of LexiconCacheMaxSizeDisk is greater than zero. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 5MB Initial value: 10MB
PromptCacheMaxSizeDisk	Maximum allowed size of the disk cache for prompt files. When the cache size exceeds this value, the least recently used prompt resources are removed from the disk cache. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 50MB Initial value: 250MB
PromptCacheMaxSizeMemory	Maximum allowed size of the memory cache for prompt files. When the cache size exceeds this value, the least recently used prompt resources are deleted from memory and serialized to the disk cache. Include a unit identifier of B (bytes), KB (kilobytes), MB (megabytes), or GB (gigabytes) with the value. If you do not include a unit identifier, the default unit is B. Default: 10MB Initial value: 25MB

Setting	Description
ResourceManagerHeuristicFraction	Value of the heuristic fraction used when calculating a value for the max-age of cached resources, where one of the following HTTP header values is set: • no expires • cache-control: max-age • cache-control: s-maxage The algorithm used is described in the HTTP 1.1 specification (RFC 2616). Valid values: 0.0 - 1.0 Default: 0.1
ResourceManagerHeuristicMaximum	Value of the heuristic maximum used in the algorithm described in ResourceManagerHeuristicFraction. Valid values: 0 - ***n*** Default: 86400

SSML Processor media engine settings

Use the following settings in the SSML Processor configuration file (*ssmlprocessor.conf)* to configure settings for the SSML Processor media engine:

Setting	Description
RTSPSharingEnabled	Specifies whether multiple clients can attach to the same RTSP session. If true, a client can attach to an already playing presentation without having to start a new RTSP session with the streaming server, provided that both of the following are true: • The audio/video codecs requested by the client match those in which the presentation is already being delivered. • The presentation is live. Pre-recorded presentations are shared only if the new client attaches to the existing session while it is setting up the media streams, but before playback has actually begun. This RTSP sharing feature is useful in situations where streaming server licenses are limited. Valid values: • true: Allows multiple clients to attach to the same RTSP session. • false: Disables the RTSP sharing feature for all requests. Default: true To disable RTSP sharing on a request-by-request basis, append ;noshare to the RTSP URI specified in the <audio> element. For example, `<audio src=”rtsp://10.0.0.1:554/live.sdp;noshare”/>.`

Setting	Description
RTPIdleTimeoutMS	Maximum idle time in ms for the RTSP client to wait at the end of playback before it ends the session. (If no RTP packets are received during the specified time, the presentation is deemed to have finished.) The RTP idle timeout feature is relevant mainly to live presentations, where the presentation length cannot be determined in advance. Valid values:0 - 3600000 Default: 2000 (2 seconds)
RTPPortRangeStart	Lowest UDP port to be used for RTP streams. This setting is used for both the media engine and the MRCP engine. Valid values: 0 - 4000 Default: 4000.
RTPPortRangeEnd	Highest UDP port to be used for RTP streams. This setting is used for both the media engine and the MRCP engine. Valid values: 0 - 9000 Default: 9000

SSML Processor MRCP settings

Use the following settings in the *ssmlprocessor.conf* file to enable the SSML Processor to use one or more MRCP text-to-speech (TTS) engines.

For information about configuring MRCP TTS resources, see *Configuring MRCP TTS resources* on page 54.

Setting	Description
MRCPConfig	Location of the MRCP configuration file, an XML file that specifies information about available MRCP TTS servers. Default: *vx/ssmlprocessor/conf/mrcp.xml*
MRCPConnectTimeoutMS	Maximum time in ms to wait for a response from an MRCP server. Valid values: 0 - 3600000 Default: 5000 (5 seconds)
MRCPTTSRTPidleTimeoutMS	Maximum time period in ms for which an MRCP TTS client waits for RTP packets to be received, before the client tears down the session. Valid values: 0 - 3600000 Default: 2000 (2 seconds)
RTPPortRangeEnd	Highest UDP port to be used for RTP streams. This setting is used for VX Integrated Media Platform media engine and MRCP engines. Valid values: 0 - 9000 Default: 9000
RTPPortRangeStart	Lowest UDP port to be used for RTP streams. This setting is used for the VX Integrated Media Platform media engine and MRCP engines. Valid values: 0 - 4000 Default: 4000.

5 Configuring the Media Resource Function

Overview of configuring the Media Resource Function

The Media Resource Function of the VoiceXML Subsystem is implemented by the following components:

- SIP media interface object (MIOSIP)
- Audio driver (ADTCP) library

MIOSIP

MIOSIP provides:

- SIP call control
- Media processing over RTP
- DTMF generation and recognition (through SRGS)
- An MRCP client to automatic speech recognition (ASR) engines

The VoiceXML Interpreter loads MIOSIP at startup.

MIOSIP uses the Vision™ Media Engine for playback. You can optionally modify the default settings for the Vision™ Media Engine in the SSML configuration file (*ssmlprocessor.conf*). For information, see *SSML Processor media engine settings* on page 36.

The engine configuration file, *config.xml*, reserves the resources needed to use the Vision media engine. Do not change values for the first occurrence of <engine>, <locale>, <server>, and <vendor> in this file, since these settings are pre-set for the Vision media engine.

The DefaultStream parameter specifies how audio files fetched by HTTP are streamed to the client. By default, audio files in a supported format are streamed to the client as they are being fetched. Because the media engine does not buffer the audio data, the resulting audio can be disjointed if bandwidth is limited.

To force the media engine to wait until the entire file has been fetched before rendering it, set the value of the DefaultStream parameter to false. To override the default behavior on a case-by-case basis, use the audio element's stream parameter. For example:

```
<audio src=http://"acme.com/audiofile.wav;stream=false"/>.
```

The following example shows the pre-set Vision™ media engine settings in bold text:

```
<?xml version="1.0"?>
<config>
    <engine id="localhost">
        <locale name="MediaEngine"/>
        <server address="localhost"/>
        <vendor name="Vision MediaEngine"/>
        <instances number="5"/>
        <parameter name="DefaultStream" value="true"/>
    </engine>
</config>
```

ADTCP library

ADTCP is the audio driver that provides a TCP interface to MIOSIP for rendering SSML fragments. The SSML Processor loads the ADTCP library at startup.

For more information about ADTCP, see *Configuring ADTCP* on page 50.

Configuring MIOSIP

To instruct the VoiceXML Interpreter to activate MIOSIP at startup, leave the default value for the MIOImplName configuration setting in the *vxmlinterpreter.conf* file as follows:

```
MIOImplName=miosip
```

Pass settings to MIOSIP through the MIOSIPImplParams configuration parameter, which is a space-separated list of ***setting-name***=***setting-value*** pairs.

If ***setting-value*** is a long string, spread the value over several lines by placing a backslash (\) at the end of every line. Separate each name-value pair with spaces.

This topic provides information about:

- Supported formats
- Dialog configuration settings
- Media configuration settings
- DTMF configuration settings
- MIOSIP logging
- MIOSIP resource selection

Supported formats

MIOSIP supports a number of audio file formats and codecs including those required by SSML and several others. It uses the SoX (Sound eXchange) audio conversion library to perform conversion between formats.

The following formats are required by the SSML specification:

- Raw (headerless) 8 kHz 8-bit mu-law (PCM) single channel (G.711)
- Raw (headerless) 8 kHz 8-bit A-law (PCM) single channel (G.711)
- WAV (RIFF header) 8 kHz 8-bit mu-law (PCM) single channel
- WAV (RIFF header) 8 kHz 8-bit A-law (PCM) single channel

Additional formats are:

- WAV (RIFF header) 8 kHz 16-bit linear (PCM) single channel
- Sun audio (AU) 8 kHz 8-bit mu-law (PCM) single channel
- Sun audio (AU) 8 kHz 8-bit A-law (PCM) single channel

- Sun audio (AU) 8 kHz 16-bit linear (PCM) single channel
- SPHERE (NIST header) 8 kHz 8-bit mu-law (PCM) single channel
- SPHERE (NIST header) 8 kHz 16-bit linear (PCM) single channel
- 3GP multimedia files containing H.263-encoded, H.264-encoded, or MPEG4-encoded video; and AMR NB-encoded audio. Dialogic recommends using a default rate of 12.2 kbit/s on the audio stream.

In general, if a file fetched through HTTP is in a supported format, the media engine converts it to the format required by the SSML Processor before it is played. For audio files, this format is RAW, 8-bit, 8 kHz, single channel, mu-law or A-Law, depending on which audio codec the client issuing the SSML request requests. If the file is eligible for caching, the cache is then updated with the converted version. This avoids having to re-convert the file if a subsequent SSML document requests this file and the cached version is eligible for use.

Certain audio file formats can be streamed directly to the client while the audio data is received from the remote server. For large files that can take a long time to fetch completely, streaming directly to the client helps avoid a noticeable latency. Such formats must include a recognizable header and be 8-bit, 8 kHz, single channel, mu-law or A-law.

Currently, the only format supported for streaming is the Microsoft® RIFF format. Once the header is processed, the audio data is streamed to the client as the chunks of data arrive. When the entire file is received, and if it is eligible for caching, it is converted and added to the cache so that subsequent requests can avail themselves of the cached resource without having to convert it. If the file does not include a suitable header, it is not streamed, but converted and played after the entire fetch has completed.

Dialog configuration settings

The following table describes the dialog parameters you can pass to the Media Resource Function from the MIOImplParams configuration parameter in the MIOImplParams configuration parameter in the *vxmlinterpreter.conf* file:

Setting	Description
com.vision.miosip.dialog.FromToUseEntireSIPURI	Creates backwards compatibility in a VoiceXML application that requires the user part only from the From and To fields of the SIP INVITE. Valid values: • true: Use the full SIP URIs from the From and To fields of the SIP INVITE when setting the corresponding VoiceXML session variables. • false: Use the user part only from the From and To fields of the SIP INVITE when setting the corresponding VoiceXML session variables. Default: true (no backwards compatibility)

Setting	Description
com.vision.miosip.dialog.TelHosts	Address or addresses of SIP User Agents to use for outbound or transfer calls when the tel: URI syntax (RFC 2806) is used to specify the destination number. The value is a comma-separated list of SIP addresses, specified as *ip4_address:port* or a fully qualified domain name. You can associate a priority with each address by delimiting with [**x**] where **x** is the priority. Lower values are selected first. Addresses with the same priority values are selected randomly. The default priority value is 0. In the event of a failure, the next address is tried according to these rules, until the list is exhausted. If a fully qualified domain name is specified, DNS SRV records are used and the behavior of RFC 3263 applies. Default: *localhost*
com.vision.miosip.dialog.Transport	SIP transport type to use. Valid values: • udp • tcp Default: tcp

com.vision.miosip.rvsip.authentication	Lists supported realm, user name, and password combinations for authenticating the VoiceXML Interpreter to the SIP location server. When the SIP location server wants to authenticate the VoiceXML Interpreter, it generates a digest challenge that contains a realm parameter. The VoiceXML Interpreter responds to the challenge with the appropriate user name and password for the specified realm.
	The format is:
	realm:***user-name*:*password***[,***realm*:*user-name*:*password***]*
	For example:
	`vision.com:vision-user:vision-pasword,realm.proxy.com:userA:passwordA`
	To specify a default user name and password for all unknown realms, use a colon in place of a realm name:
	`:default-user:default-password`

Setting	Description
com.vision.miosip.rvsip.defaultLogFilters	Default logging level for all SIP stack modules. Use the logical-OR (|) to combine any number of the following values: • RVSIP_LOG_DEBUG_FILTER • RVSIP_LOG_INFO_FILTER • RVSIP_LOG_WARN_FILTER • RVSIP_LOG_ERROR_FILTER • RVSIP_LOG_EXCEP_FILTER • RVSIP_LOG_ENTER_FILTER • RVSIP_LOG_LEAVE_FILTER • RVSIP_LOG_LOCKDBG_FILTER Default: all (full logging)
com.vision.miosip.rvsip.dnsServers	Comma-delimited list of DNS servers to use for outbound SIP messages. If no servers are listed, MIOSIP uses the operating system default.
com.vision.miosip.rvsip.ePersistencyLevel	Persistency level used by the SIP stack objects. Valid values: RVSIP_TRANSPORT_PERSISTENCY_LEVEL_UNDEFINED SIP stack objects always open a new connection for sending requests. Responses are sent on the connection on which received. RVSIP_TRANSPORT_PERSISTENCY_LEVEL_TRANSC A SIP stack transaction object sending a request uses a connection from the hash, if a suitable connection is found. If the SIP stack must create a new connection, it adds it to the hash. The transaction object only detaches from the connection when it terminates. RVSIP_TRANSPORT_PERSISTENCY_LEVEL_TRANSC_USER Same as RVSIP_TRANSPORT_PERSISTENCY_LEVEL_TRANSC with the added rule that call legs always try to use the same connection for all outgoing requests. Default: RVSIP_TRANSPORT_PERSISTENCY_LEVEL_TRANSC_USER
com.vision.miosip.rvsip.localTcpPort	Local TCP port on which the SIP stack listens. Default: 5060
com.vision.miosip.rvsip.localUdpPort	Local UDP port on which the SIP stack listens. Default: 5060
com.vision.miosip.rvsip.locationServerAddress	Address of a SIP location server URI with which the Media Resource Function registers. Can be an IP4 address or a fully qualified domain name, in which case RFC 3263 behavior applies. There is no default address.

Setting	Description
com.vision.miosip.rvsip.logContext	Path and file name for the SIP stack used by the VoiceXML Subsystem. For example, *vx/vxmlinterpreter/logs/rvsip.log*. Default: Blank (no log file)
com.vision.miosip.rvsip.maxCallLegs	Maximum number of call legs the SIP stack allocates. Set this value to the maximum number of calls you expect the SIP stack to handle simultaneously. Default: 240
com.vision.miosip.rvsip.outboundProxyHost	Address of the SIP proxy through which outbound SIP messages are passed. Can be an IP4 address or a fully qualified domain name, in which case RFC 3263 behavior applies. There is no default SIP proxy server.
com.vision.miosip.rvsip.outboundProxyPort	SIP port on the SIP proxy server. Default: 5060
com.vision.miosip.rvsip.outboundProxyTransport	Transport type to use with the SIP proxy server. Valid values: • udp • tcp Default: tcp
com.vision.miosip.rvsip.registerRetryPeriod	Interval, in seconds, with which the VoiceXML Interpreter tries to register with the SIP location server. Valid values: 60 - ***n*** Default: 1800
com.vision.miosip.rvsip.sendReceiveBufferSize	Maximum size of SIP messages, in bytes. Default: 30240.
com.vision.miosip.rvsip.userAgentAoR	Public URI to pass to the SIP location server, specified as a valid SIP URI. Default: *sip:user@miosip.vision.com*
com.vision.miosip.rvsip.userAgentContact	Actual contact URI to pass to the SIP location server, specified as a valid SIP URI.

Media configuration settings

The following table describes the media setting parameters you can pass to the Media Resource Function from the MIOImplParams configuration parameter in the *vxmlinterpreter.conf* file:

Setting	Description
com.vision.miosip.defaultTransferAudio	Path of a local file used as the transferaudio while connecting a tromboned transfer. This must be a raw G7.11 audio file. Default: *vx/vxmlinterpreter/data/miosip/transferaudio.ulaw*

Setting	Description
com.vision.miosip.media.AMRFormat	Format of the AMR audio stream to be used in video calls. Valid values: • rfc3267: RFC 3267 octet-aligned AMR stream • if2: 3GPP 26.101 Interface Format 2 Default: rfc3267
com.vision.miosip.media.ConnectTimeoutMS	Generic TCP connection timeout value in milliseconds. Default: 2000
com.vision.miosip.mediacontrol	Protocol used to communicate with media resources such as automatic speech recognition (ASR) servers. Valid values: • mrcp • ivre Default: mrcp
com.vision.miosip.media.AudioCallPreferredAudioCodec	Preferred audio codec when offering media to a remote user agent for an audio call (that is, for late-offer or outbound call legs). Valid values: • g711-ulaw • g711-alaw Default: g711-alaw
com.vision.miosip.media.DefaultAsrEngine	Default ASR engine vendor to use when there are two or more engines in the MRCP configuration file that match the language in question. This can be overridden at runtime by the VoiceXML <property> com.vision.asrengine. The default is "" (no vendor). Therefore, if the VoiceXML <property> is not set, the first engine in the MRCP configuration file that matches the active language is used.
com.vision.miosip.media.ForceComfortNoise	Indicates whether the Media Resource Function uses comfort noise packets (RTP payload type 13) during silent periods in audio calls, despite the SDP offer at call startup. Valid values: • true: Media Resource Function uses comfort noise packets during silent periods. • false: Media Resource Function does not use comfort noise packets during silent periods. Default: true
com.vision.miosip.media.MRCPConfigFile	Path of the VoiceXML Interpreter MRCP XML configuration file. Default: *vx/vxmlinterpreter/conf/mrcp.xml*

Setting	Description
com.vision.miosip.media.RecordedAudioHTTPRoot	HTTP path from which recorded audio is accessible via HTTP GET. Default: *http://<local_address>:<local_port>/audio/*
com.vision.miosip.media.RecordedAudioLocation	Path to which recorded audio is saved. This location must be servable by an HTTP server specified in the com.vision.miosip.media.RecordedAudioHTTPRoot setting. Default: *vx/vxmlinterpreter/www/audio/*
com.vision.miosip.media.RecordVADSensitivitySNRRange	Absolute range of signal-to-noise ratio (SNR) variance when modified by the VoiceXML sensitivity property. The formula used to calculate the actual SNR during a record action is: RecordVADThresholdSNR – ((sensitivity – 0.5) * RecordVADSensitivitySNRRange) Default: 10
com.vision.miosip.media.RecordVADThresholdSNR	Threshold signal-to-noise ratio (SNR) used by default when performing voice activity detection during a record action. This occurs when the VoiceXML sensitivity property is set to the default (0.5). Default: 14
com.vision.miosip.media.RTP4IVRImplName	Name of the RTP engine implementation to load at startup. Value must be vprtp4ivr for this release. Default: vprtp4ivr
com.vision.miosip.media.RTPInterval	Interval between RTP packets in periodic streams, in milliseconds. Default: 20
com.vision.miosip.media.RTPPortRangeEnd	Highest UDP port used for RTP streams. Default: 9000
com.vision.miosip.media.RTPPortRangeStart	Lowest UDP port used for RTP streams. Default: 4000
com.vision.miosip.media.SSMLProcessorAddresses	Comma-separated list of SSML processor IP addresses, specified as *ip4_address:port.* A priority can be associated with each IP address by delimiting with [**x**] where **x** is the priority. Lower values are selected first. Addresses with the same priority values are selected randomly. The default priority value is 0. In the event of a failure, the next address is tried according to these rules until the list is exhausted. Default: 127.0.0.1:32323
com.vision.miosip.media.stack.LogFile	Path of the MRCP log file. Default: *vx/vxmlinterpreter/logs/mrcpstack.log*

Setting	Description
com.vision.miosip.media.stack.LogFilter	Logging level for the MRCP module. Use the logical-OR (|) to combine any number of the following values: • MRCP_STACK_INFO • MRCP_STACK_MESSAGE • MRCP_STACK_TRANSPORT • MRCP_STACK_DEBUG The following example enables the MESSAGE and INFO filters. `com.vision.miosip.media.stack.LogFilter= MRCP_STACK_MESSAGE|MRCP_STACK_INFO` Default: Blank (no logging). However, WARNING and ERROR messages are always logged, regardless of which filters are specified.
com.vision.miosip.mediaTTSAllowables	Comma-separated list of languages allowed by the SSML Processor. Other languages result in a VoiceXML error.noresource event. Default: None (allows all languages)
com.vision.miosip.media.VideoCallPreferredAudioCodec	Preferred audio codec when offering media to a remote user agent for a video call (that is, for late-offer or outbound call legs). Valid values: • amr-if2 • amr-rfc3267 • g711-alaw • g711-ulaw Default: amr-rfc3267
com.vision.miosip.media.VideoCallPreferredVideoCodec	Preferred video codec when offering media to a remote user agent for a video call (that is, for late-offer or outbound call legs). Valid values: • h263-2190 • h263-2429 • h264-3984 • mpeg4-3016 • none Default: h263-2429

Setting	Description
com.vision.miosip.media.VideoCallRecordOnIframeTimeout	Specifies when the VoiceXML media server starts recording a video stream. Valid values: • -1: wait for I-frame before recording • 0: start recording immediately • >0: wait for I-frame for specified period of time (in ms) before recording Default: -1
com.vision.miosip.media.VideoCallRecordOnIframeRetry	Specifies the number of times a video fast update is attempted. Video fast update is retried until an I-frame is received or the number of retries is reached. Valid values: • -1: retry forever • 0: no retry • >0: number of times video fast update is retried if no I-frame is received in response to a video fast update. Default: 0
com.vision.miosip.media.VideoCallRecordOnIframeRetry	Delay between retries of video fast update requests (in ms)

DTMF configuration settings

Use the following settings in the MIO settings section of the v*xmlinterpreter.conf* file to determine how the Vision VoiceXML Subsystem handles inbound and outbound messages with DTMF content:

Name	Description
acceptDTMFInSIPINFO	Indicates whether the VoiceXML Interpreter can accept an incoming SIP INFO message with DTMF content. Valid values: • true: VoiceXML Interpreter can accept an incoming SIP INFO message with DTMF content when RFC 2833 is not negotiated. • false: VoiceXML Interpreter cannot accept an incoming SIP INFO message with DTMF content. It rejects the message by issuing a 415 Media not supported response. Default: false
outboundDTMFConfig	Determines how the VoiceXML Interpreter sends an outbound DTMF. Valid values: • 0: VoiceXML Interpreter always sends an outbound DTMF according to RFC 2833, if the other party supports RFC 2833. If the other party does not support RFC 2833, the VoiceXML Interpreter does not send DTMFs. • 1: VoiceXML Interpreter always sends an outbound DTMF in a SIP INFO message. • 2: VoiceXML Interpreter sends an outbound DTMF according to RFC 2833, if the other party supports RFC 2833. If the other party does not support RFC 2833, the VoiceXML Interpreter sends the DTMF in a SIP INFO message. Default: 0

MIOSIP logging

MIOSIP logging works as follows:

- MIOSIP sends log messages to the VoiceXML Interpreter's system log file. Log messages from the MIOSIP are prefixed with the expression, LOG_MESSAGE_FROM_MIO.
- If the com.vision.miosip.rvsip.logContext parameter is set, RVSIP logs all SIP stack-related messages to the specified file.
- If the com.vision.miosip.media.stack.LogFile parameter is set, all MRCP-related messages are sent to the specified file.

MIOSIP resource selection

A number of MIOSIP configuration parameters allow the specification of multiple resource addresses with an optional associated priority. These resources are specified as:

addressA:***portA***[***priorityA***],***addressB***:***portB***[***priorityB***],...

The following rules apply when a resource is required by MIOSIP:

- Start with the IP address/FQDN with the lowest priority value. A resource with no priority value assumes value 0.

- IP addresses/FQDNs with equal priorities are randomly selected.
- If a failure occurs, that IP address/FQDN is removed from the list and the remaining list is retried. The process starts at the beginning and repeats until the list is exhausted.
- Load-balancing and failover logic is per-request, for example, stateless.

Sample configuration types are shown in the following table:

Configuration type	Example
Non-redundant/standalone configuration	10.0.0.10
Load balanced configuration	10.0.0.20[0].10.0.0.21[0]
Weighted load balanced configuration	10.0.0.30[0],10.0.0.30[0],10.0.0.30[0],10.0.0.31[0]
Redundant/failover configuration	10.0.0.40[0],10.0.0.41[1]
Redundant and load balanced configuration	10.0.0.50[0],10.0.0.51[0],10.0.0.52[1],10.0.0.53[1]

Configuring ADTCP

To instruct the SSML Processor to activate ADTCP at startup, set the ADImplName configuration setting in the *ssmlprocessor.conf* file as follows:

```
ADImplName=adtcp
```

Pass settings to ADTCP through the ADImplParams configuration parameter, which is a space-separated list of ***setting-name***=***setting-value*** pairs.

The following parameters configure the behavior of ADTCP:

Setting	Description
com.vision.CharSet	Character set to use when converting incoming SSML documents to wide chars before queuing the document in the SSML processor. Default: UTF-8. **Note:** Do not change the default setting.
com.vision.TTSListenPort	Port on which to listen for SSML requests from clients.

Configuring the DTMF generator

The Vision™ VoiceXML DTMF generator is a dynamic library used to create DTMF events from VoiceXML applications. To enable the DTMF generator, pass the following setting to the Media Resource Function by using the MIOImplParams configuration parameter in the *vxmlinterpreter.conf* file:

VXMLObjects=VODTMFGenerator

For information about using the DTMF generator, see the *Dialogic® Vision™ VX Integrated Media Platform User's Manual*.

6 Configuring ASR and TTS resources

Overview of configuring ASR and TTS resources

The VoiceXML Subsystem provides the ability to use third-party ASR and TTS resources for audio and video calls. The following tables describe the supported resources:

Automatic speech recognition (ASR) engines

The VX Integrated Media Platform supports the following ASR engines through MRCP:

Vendor	Product
Loquendo	Speech Suite v7.0.9 (ASR v7.2.0)
Nuance	NRec v9.0.0 / NSS v5.0.0
	Nuance v8.5 (MRCP-1-0-0-SP10)
ScanSoft	OSR v3.0.13 / SWMS v3.1.14
Telisma	TeliSpeech 1.2 Service Pack 2

For information about configuring ASR engines, see *Configuring ASR resources* on page 52.

Text-to-speech (TTS) engines

The VX Integrated Media Platform supports the following TTS engines through MRCP:

Vendor	Product
CVOX	CVOX + France Telecom MRCP Server v1.1
Loquendo	Speech Suite v7.0.9 (TTS v6.5.6)
ScanSoft	RealSpeak Telecom 4.0.12 + SWMS v3.1.14

For more information, see *Configuring MRCP TTS resources* on page 54.

The VX Integrated Media Platform also supports the Scansoft RealSpeak Telecom 4.0.12 TTS engine through native API integration (non-MRCP). For more information, see *Configuring non-MRCP TTS resources* on page 55.

Configuring ASR resources

To access automatic speech recognition (ASR) resources through the VX Integrated Media Platform, follow these steps:

Step	Action	For more information, see...
1	In the VoiceXML Interpreter MRCP configuration file (vx/*vxmlinterpreter/conf/mrcp.xml*), configure the required settings for the third-party ASR engines you want to use.	*VoiceXML Interpreter MRCP configuration file settings* on page 52.
2	Ensure that the required ASR engines are installed correctly with the VX Integrated Media Platform.	*Third-party ASR and TTS engine notes* on page 56.
3	Set the appropriate values for the ASR-related runtime settings.	*Runtime settings for ASR* on page 53.

VoiceXML Interpreter MRCP configuration file settings

The VoiceXML Interpreter MRCP configuration file is an XML file that specifies information about the MRCP ASR engines available to the VX Integrated Media Platform. By default, this file is named *mrcp.xml*, and it resides in the vx/*vxmlinterpreter/conf* directory.

Note: Do not confuse this MRCP file with the MRCP file located in the *vx/ssmlprocessor/conf* directory. The latter file is used to specify text to speech (TTS) engines for the SSML Processor. For information, see *Configuring MRCP TTS resources* on page 54.

The VoiceXML Interpreter *mrcp.xml* file contains the <config> root element, and one <engine> child element for each ASR engine.

To define an ASR engine, specify the language, vendor, and URI for the engine. The following table describes the attributes of the resource and server elements used in the VoiceXML Interpreter MRCP configuration file:

Element	Attribute	Description
resource	lang	Comma-separated list of languages supported by this resource. A wild card ("*") means that all languages are allowed.
	resourcetype	MRCP resource types for speech recognition. Only valid value is speechrecog.
	vendor	Vendor associated with this resource. Valid values are: • Loquendo • Nuance • Scansoft • Telisma
	persist	Boolean value specifying if the media session should be kept open between media processing requests as follows: • true: MIOSIP connects to this resource upon the first request and keeps the connection open until the call ends. • false: MIOSIP disconnects after each request.
server	uri	RTSP URI for this resource.
	priority	Priority of this ASR engine over other ASR engines specified with the same resource. ASR engines defined with the lower priority are used first. ASR engines defined with equal priorities are selected at random. In the event of a failure, the next ASR engine is tried according to these rules, until the set of ASR engines specified by the <server> element is exhausted. Default: 0.

The following example defines a ScanSoft ASR engine that supports the Great Britain variant of English:

```
<mrcp-config>
  <resource lang="en-GB" resourcetype="speechrecog" vendor="ScanSoft" persist="true">
    <server uri="rtsp://10.10.35.87:4900/media/speechrecognizer" priority="0" />
  </resource>
</mrcp-config>
```

Runtime settings for ASR

The VoiceXML Interpreter com.vision.asrengine runtime setting specifies which ASR engine vendor to use when there are two or more ASR engines in the MRCP configuration file that match the specified language. The value of com.vision.asrengine overrides the value of com.vision.miosip.media.DefaultAsrEngine, which is specified by the MIOImplParams parameter in the *vxmlinterpreter.conf* file.

If both of these settings are blank, the VX Integrated Media Platform uses the first ASR engine in the VoiceXML Interpreter MRCP configuration file (vx/*vxmlinterpreter/conf/mrcp.xml*) that matches the specified language.

For more information about the com.vision.asrengine setting, see *VoiceXML Interpreter runtime settings* on page 27. For more information about MIOImplParams, see *Configuring MIOSIP* on page 40.

Configuring MRCP TTS resources

The easiest way to access TTS resources through the VoiceXML engine is to use MRCP TTS engines for speech synthesis. To use MRCP TTS engines, follow these steps:

Step	Action	For more information, see...
1	In the SSML configuration file (*vx/ssmlprocessor/conf/ssmlprocessor.conf*), set appropriate values for the MRCP initialization settings.	*SSML Processor MRCP settings* on page 37.
2	In the engine configuration file (*vx/ssmlprocessor/conf/config.xml*), reserve the resources needed to use MRCP TTS engines.	*Reserving resources for MRCP* on page 54.
3	In the SSML Processor MRCP configuration file (vx/*ssmlprocessor/conf/mrcp.xml*), configure the required settings for the third-party MRCP TTS engines you want to use.	*SSML Processor MRCP configuration file settings* on page 54.
4	Ensure that the required MRCP TTS engines are installed correctly with the VoiceXML engine.	*Third-party ASR and TTS engine notes* on page 56.

For information about supported MRCP TTS engines, see *Text-to-speech (TTS) engines* on page 51. For information about the resource selection process the SSML Processor uses when a client requests an MRCP resource, see *MIOSIP resource selection* on page 49.

Reserving resources for MRCP

To reserve resources for MRCP, set the <locale> name attribute to "*" and the <vendor> name attribute to MRCPEngine in the engine configuration file (*config.xml*). The following example reserves the resources needed to support two ports. This gives two calls simultaneous access to MRCP resources.

```
<engine id="mr01">
    <locale name="*" />
    <server address="localhost" />
    <vendor name="MRCPEngine" />
    <instances number="2" />
</engine>
```

SSML Processor MRCP configuration file settings

By default, the SSML Processor MRCP configuration file is named *mrcp.xml*, and it resides in the vx/*ssmlprocessor/conf* directory. This file contains the <config> root element, and one <engine> child element for each media or TTS engine.

Note: Do not confuse this MRCP file with the MRCP file located in the *vx/vxmlinterpreter/conf* directory. The latter file is used to specify ASR engines for MIOSIP. For information, see *Configuring ASR resources* on page 52.

The following table lists the XML elements that configure each MRCP TTS engine:

Element	Attribute	Description
<resource>	lang	Comma-separated list of languages supported by this resource. A wildcard (“*”) means that all languages are allowed.
	type	MRCP resource type for this resource. Must be speechsynth.
	vendor	Vendor associated with this resource. Valid values are: • CVOX • Loquendo • ScanSoft
	persist	Boolean value specifying if the media session should be kept open between media processing requests.
<server>	uri	RTSP URI for this resource.
	priority	Priority of this engine over other engines specified within the same resource. Engines with the lower priority are always used first. Engines with equal priority are selected at random. Default priority is 0. In the event of a failure, the next engine is tried according to these rules until the set of engines is exhausted.

The following example defines a ScanSoft ASR engine that supports the Great Britain variant of English.

```
<mrcp-config>
<resource lang="en-GB" resourcetype="speechsynth" vendor="ScanSoft" persist="true">
 <server uri="rtsp://127.0.0.1:4900/media/speechsynthesizer" priority=”0”/>
</resource>
</mrcp-config>
```

Configuring non-MRCP TTS resources

The Scansoft RealSpeak Telecom 4.0.12 TTS engine is also available with native API integration. To use a ScanSoft RealSpeak engine with native API integration, follow these steps:

Step	Action
1	Ensure that the required TTS server is installed correctly with the VX Integrated Media Platform.
2	Make the following changes to the engine configuration file, *vx/ssmlprocessor/conf/config.xml*: • Set the <vendor> name attribute to ScanSoft RealSpeak 4. • Set the <parameter> name attribute to language and the associated <parameter> value attribute to the RealSpeak-specific language string used by the engine to determine an internal locale setting. For example, British English. • Set another <parameter> name attribute to voice and the associated <parameter> value attribute to the RealSpeak-specific name of the voice to be used. For example, Emily. • Set the <parameter> name to dictionarypath and the associated value attribute to the location of a ScanSoft-specific user-dictionary.
3	To use ScanSoft RealSpeak 4 with the SSML Processor, you must also install the RealSpeak 4 SDK on the machine on which the SSML Processor runs. For more information, see *Scansoft* on page 58.

For general information about ScanSoft RealSpeak, see the ScanSoft RealSpeak documentation.

The following example supports 20 simultaneous connections to the RealSpeak 4 TTS Server. The TTS server is installed on the same machine as the VX Integrated Media Platform, and it supports the en-GB language with the Emily voice:

```
<?xml version="1.0"?>
<config>

    <!-- Engine for handling audio files -->
    <engine id="ae01">
        <locale name="MediaEngine"/>
        <server address="localhost"/>
        <vendor name="vision MediaEngine"/>
        <instances number="144"/>
        <parameter name="DefaultStream" value="true"/>
    </engine>

    <!-- RealSpeak 4.0 -->

    <engine id="TTS-Realspeak-20">
        <locale name="en-GB"/>
        <server address="10.10.35.87:6666"/>
        <vendor name="ScanSoft RealSpeak 4"/>
        <instances number="20"/>
        <parameter name="language" value="British English"/>
        <parameter name="voice" value="Emily"/>
        <parameter name="dictionarypath" value="%visionvx%\ssmlprocessor\dict\en-
GB\dict.txt"/>
        <parameter name="gender" value="female"/>
    </engine>

</config>
```

Third-party ASR and TTS engine notes

You must take additional steps to enable third party ASR and TTS engines to operate correctly within the VX Integrated Media Platform. In many cases, the information is specific to the current version of the third party engine in question; for example, it may refer to a bug in the current version and describe a work-around for the bug.

Note: This information might change as third party engines are upgraded in future releases of the VX Integrated Media Platform.

To enable third party TTS and ASR resources to operate with the VX Integrated Media Platform, you must also specify the proper settings for them in the engine configuration file (*config.xml*). For information, see *Configuring MRCP TTS resources* on page 54 and *Configuring non-MRCP TTS resources* on page 55.

Nuance MRCP Server 8.5

The default installation for the Nuance MRCP Server 8.5 does not work with the VX Integrated Media Platform. Follow these steps to use Nuance MRCP Server 8.5 with the VX Integrated Media Platform:

Step	Action
1	Complete the default installation for the Nuance MRCP Server.
2	Install any required language packs, as described in the Nuance documentation.

3	Add Nuance to the SSML Processor MRCP configuration file (*vx/ssmlprocessor/conf/mrcp.xml*). For example:
	`mrcp-config>`
	`    <resource lang="en-US" resourcetype="speechrecog"`
	`        vendor="Nuance" persist="true">`
	`     <server uri="rtsp://<nuance-server-ip>:554/recognizer"/>`
	`    </resource>`
	`</mrcp-config>`
	You cannot use "127.0.0.1" or "localhost" for the value of <nuance-server-ip>. If you have to install Nuance on the same machine as there VX Integrated Media Platform, use the IP address of the machine instead of the loopback IP address.
	This issue was eliminated in Nuance SP9.

4	Add com.vision.miosip.media.DefaultAsrEngine=Nuance to the VoiceXML Interpreter configuration file (*vx/vxmlinterpreter/conf/vxmlinterpreter.conf*).
5	Add audio.rtp.LocalIPAddress=<nuance-server-ip> to the Nuance MRCP server command line. Optionally, place this setting in the watcher startup. If you leave out this setting, Nuance never receives an RTP stream with the audio, and the software throws no-inputs when you try to do speech recognition. This issue was fixed in Nuance MRCP Server SP9.
6	Add audio.rtp.PlaySilence=FALSE to the Nuance MRCP server command line. This eliminates a constant stream of silence RTP packets from the Nuance MRCP server, which results in saved bandwidth.

After you have completed these steps, the MRCP server line in the Nuance watcher startup might read as follows:

```
mrcp-server -cfg MRCP/mrcp-config
-package MRCP/mrcp-nl-nl
-package MRCP/mrcp-en-gb
-config MRCP/nuance-resources.txt
config.ClapiConfigFile=MRCP/clapi_cfg.xml
audio.rtp.PlaySilence=FALSE
audio.rtp.LocalIPAddress=10.0.0.5
```

Scansoft

<table>
<tr><th>Component</th><th>Version</th><th>Notes</th></tr>
<tr><td>RealSpeak TTS 4.0x</td><td>4.0.x</td><td>
The following notes apply to non-MRCP implementations of Scansoft.
RealSpeak 4
Follow these steps to use the ScanSoft RealSpeak 4 TTS engine with the SSML Processor:
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Install the RealSpeak 4 SDK on the machine where the SSML Processor will run.</td></tr>
<tr><td>2</td><td>Explicitly set the environment variable $SSFTTTSSDK to the RealSpeak install directory. By default, this directory is /usr/local/ScanSoft/RealSpeak_4.0/</td></tr>
</table>
For more information, see the ScanSoft RealSpeak documentation.
Dutch voice Claire
The RealSpeak documentation states that the language identifier for the Dutch voice Claire is Netherlands Dutch. However, for RealSpeak version 4.0.5, specifying this value in the SSML Processor XML configuration file does not work. Instead, set the language to Dutch in the SSML Processor XML configuration file.
</td></tr>
<tr><td>OSR ASR</td><td>3.0.3</td><td>
Warnings similar to the following example appear in the ScanSoft OSR 3.03 logs:
<pre>** WARNING **| -2| SWI_ERROR| recoverable error|
SWIepDetectorCreateInternal |increment SWIepCurrChanId = 1</pre>
This warning indicates that the channel ID of the end-pointer was incremented, and is for engineering purposes only. You can ignore this message, because there is no problem with the end-pointer when the message is written.
</td></tr>
<tr><td>Speech Works Media Server for MRCP</td><td>3.1.9</td><td>
If a large number of concurrent channels are active and using RealSpeak TTS via MRCP on the VX Integrated Media Platform, Dialogic suggests that you change the configuration for the SWMS OSSServer. The configuration file for OSSServer is called OSSserver.cfg. It resides in the SWMS_installation\config directory, where SWMS_installation is the installation directory for the SWMS OSSServer.
The following table lists the recommended values for SWMS OSSServer configuration settings if the VX Integrated Media Platform is using 120 channels of MRCP TTS. These settings are commented out by default.
<table>
<tr><th>Setting</th><th>Recommended value</th></tr>
<tr><td>server.realspeak4.audiothreadnumber</td><td>150</td></tr>
<tr><td>server.realspeak4.cache.enable</td><td>1</td></tr>
<tr><td>server.realspeak4.cache.initialNumber</td><td>120</td></tr>
<tr><td>server.realspeak4.cache.maxNumber</td><td>130</td></tr>
<tr><td>server.realspeak4.cache.timeoutSec</td><td>600</td></tr>
</table>
</td></tr>
</table>

7 Glossary

A

ADTCP: An audio driver that provides a TCP interface to MIOSIP for rendering SSML fragments.

AMR: Adaptive multi-rate; an audio data compression scheme optimized for speech coding. This scheme was adopted by 3GPP and is used in video services.

ASR: Automatic speech recognition; ASR resources, called ASR engines in the MRCP framework, typically enable users of information systems to speak entries rather than punching numbers on a keypad. See also MRCP.

Authorization and Usage Indication interface: XML-over-HTTP mechanism that authorizes call sessions and gathers information for call detail reports.

B

blind transfer: A call transfer in which the originating caller is not announced and is connected directly to destination. In a blind transfer the Vision™ Server redirects the caller to the callee without remaining in the connection and does not monitor the outcome.

bridge transfer: A blind transfer in which the Vision™ Server redirects the caller to the callee and remains as a listener.

C

Call Server: Component of the Vision™ Server that manages call control and routing capabilities.

CallPlacer interface: XML-over-HTTP mechanism for initiating outbound sessions or calls for VoiceXML applications.

CCXML: Call Control Extensible Markup Language; a W3C Working Draft standard language for providing telephony call control support for dialog systems, gateways, and conferencing services.

CCXML application definition file: A file that maps individual CCXML applications to number ranges that trigger the execution of those applications.

clock: A periodic reference signal used for synchronization on a transmission facility, such as a telephony bus. See also clock master, clock slave, clock fallback.

clock master: A board that drives the clock signal for a system of boards connected by a bus cable. See also clock slave.

clock slave: A board that derives its clock signal from a bus cable; the clock signal is driven by the bus clock master. See also clock master.

consultation transfer: A call transfer in which the Vision™ Server initiates a transfer between two parties, but does not stay attached to the call once it is successfully established. The caller remains connected to the Vision™ Server if the transfer fails.

D

DTMF: Dual tone multi frequency; an inband signaling system that uses two simultaneous voiceband tones for dialing. Also called touchtone. Some times DMTF is used to generally describe any telephony keypad press, even if tones are not generated.

G

G.711: An ITU PCM encoder/decoder specification for mu-law and A-law encoding.

H

H.100 bus: A TDM telephony bus standard for integrating hardware from various PC board vendors. The H.100 specification defines a ribbon cable bus that transports telephony voice data and signaling data across PCI boards. The H.100 bus is an interoperable superset of the H-MVIP and MVIP-90 telephony buses.

H.223: A protocol used to multiplex control and audio and video media on and off of a single DS0 within a trunk.

H.263: An ITU video compression standard. H.263 supports CIF, QCIF, SQCIF, 4CIF and 16CIF resolutions.

H.264: An ITU and ISO video compression standard that compresses video into lower bandwidth compared to H.263 and MPEG-4. H.264 is also called MPEG-4 Part 10.

I

INAP: Intelligent Network Application Part; an SS7 protocol that facilitates building platform-independent, transport-independent, and vendor-independent applications. Such applications include service switching points (SSPs), internet protocol (IP) applications, service control points (SCPs), enhanced services platforms, service circuit nodes, and other custom applications.

ISDN: Integrated services digital network; a standard for providing voice and data telephone service with all digital transmission and message-based signaling.

ISUP: ISDN user part; the SS7 protocol layer that allows for the establishment, supervision, and clearing of circuit-switched connections between two SS7 signaling points, such as central office switches. Despite its name, the ISUP layer is not unique to interconnecting. It is used to manage all types of circuit-switched connections.

ITU: International Telecommunications Union; an international standards body for telecommunications.

IVR: Interactive voice response; a telephony application in which callers interact with programs using recorded or synthesized voice prompts, DTMF digits, or speech recognition to query or deliver information.

M

Media Resource Function: Component of the Dialogic® Vision™ VX Integrated Media Platform that provides media processing including record, playback, and interfaces to speech recognition resources. The Media Resource Function is implemented by MIOSIP.

MIB: Management information base; an SNMP collection of objects that represent a managed node. Physically, a list of variables. Logically, a table with rows of variables.

MIOSIP: Implements the Media Resource Function of the Dialogic® Vision™ VX Integrated Media Platform. MIOSIP provides SIP call control, media processing over RTP, DTMF generation and recognition, and an MRCP client to automatic speech recognition (ASR) resources.

MPEG-4: An ISO/IEC standard for compressing multimedia data (video, audio, and speech).

MRCP: Media Resource Control Protocol; an application protocol for implementing automatic speech recognition (ASR) and text-to-speech services (TTS). MRCP provides a distributed system of ASR and TTS engines connected over an IP network.

MTP: Message transfer part; the SS7 protocol layers responsible for the reliable, in-sequence delivery of packets between two SS7 signaling points. The MTP functions include message routing, signaling link management, signaling route management, and congestion control.

MVIP-95: Device driver specification for H-MVIP, H.100, and H.110 telephony buses.

N

NETANN: Basic Network Media Services with SIP; an interface that enables applications in a SIP network to locate and invoke basic services on a media server. These services include network announcements, user interaction, and conferencing services. Also called RFC 4240.

O

OSP: Open Settlement Protocol; a European Telecommunications Standards Institute (ESTI) protocol used to exchange authorization, accounting, and usage information for IP telephony.

P

PSTN: Public switched telephone network; a public telephone network.

R

route: A connection path. On the PSTN network, a route is a logical collection of trunks. On the IP network, a route is a destination URL.

RTP: Real time transport protocol; a layer added to the internet protocol (IP) that addressed problems caused when real-time interactive exchanges (such as audio data) are conducted over lines designed to carry packet-switched (connectionless) data.

S

SCCP: Signaling connection control part; an SS7 protocol that provides both connection-oriented and connectionless data transfer over an SS7 network. It extends the service provided by the SS7 MTP layers by adding extended addressing capabilities and multiple classes of service. The SCCP addressing capabilities allow a message to be addressed to an individual application or database within a signaling point. See also SS7.

SDP: Session description protocol, a protocol that defines a text-based format for describing streaming media sessions and multicast transmissions.

Signaling Server: An optional component of the Vision™ Server that provides redundant and scalable ISUP signaling.

SIP: Session initiation protocol. An IP signaling and telephony control protocol used mainly for voice over IP calls and multimedia communications. SIP relies on the session description protocol (SDP) for session description and the Real Time Transport Protocol (RTP) for actual transport.

SRGS: Speech Recognition Grammar Specification (SRGS); a syntax for representing the grammars used in speech recognition.

SS7: Signaling system 7; an out-of-band signaling system that provides fast call setup using circuit-switched connections and transaction capabilities for remote database interactions.

SSML: Speech Synthesis Markup Language; a proposed standard for enabling access to the internet using speech. SSML provides a standard way to control various aspects of speech (such as pronunciation, volume, pitch, and rate) over a variety of platforms.

SSML Processor: Component of the Dialogic® Vision™ VX Integrated Media Platform that processes SSML requests for audio and text-to-speech.

T

T.38 fax: A standard for real-time fax over IP that makes it possible for fax machines from different vendors to talk to each other over IP networks. The T.38 standard defines how to conduct group 3 facsimile transmission between terminals in which a portion of the transmission path between terminals includes (besides the PSTN or ISDN) an IP network such as the internet.

TCAP: Transaction capabilities application part; an SS7 protocol that provides applications with transaction support over the SS7 network. It enables the exchange of non-circuit related data, such as database queries and responses and remote feature invocation requests between SS7 signaling points. The TCAP layer relies on both the MTP and SCCP layers for message addressing and delivery.

TDM: Time division multiplexing; a technique for transmitting a number of separate data, voice, or video signals simultaneously over one communications medium by quickly interleaving a piece of each signal one after another.

telecom configuration file: File that provides information about the resources that interface with the Call Server and about other elements, such as the number of routes and the circuit selection.

trunk: The physical interface between the telephone network and the Vision™ Server. In telephone networks, a trunk is a shared connection between two switches. It differs from a line in that it is not dedicated to one subscriber or extension. T1 and E1 trunks carry 24 and 31 circuits, respectively.

TTS: Text-to-speech; a system that converts written language to speech.

V

Vision™ Console: Web-based configuration tool that configures the Vision™ Server.

VoiceXML: Voice Extensible Markup Language; a language that enables users to interact with the internet through voice recognition technology.

VoiceXML application configuration file: A file that maps individual VoiceXML applications to number ranges that trigger the execution of those applications.

VoiceXML Interpreter: Component of the Dialogic® Vision™ VX Integrated Media Platform that interprets VoiceXML dialogs.

VoiceXML Subsystem: Component of the Dialogic® Vision™ VX Integrated Media Platform that provides media processing for VoiceXML applications. The VoiceXML Subsystem consists of the VoiceXML Interpreter, SSML Processor, and Media Resource Function.

Index