

VOICEGENIE

VoiceGenie 7
TDD/TTY Users' Guide

September 12th, 2005



VoiceGenie Technologies Inc.
1120 Finch Ave. W. • Toronto, Ontario • M3J 3H7 • Canada
T. +1.416.736.4151 • F. +1.416.736.1551 • support@voicegenie.com
www.voicegenie.com

VoiceGenie Contacts

VoiceGenie Technologies Inc.
1120 Finch Avenue West
Toronto, Ontario
Canada
M3J 3H7

T. +1.416.736.4151
F. +1.416.736.1551
support@voicegenie.com

<http://www.voicegenie.com/index.html>

Proprietary / Copyright Information

This material contains proprietary and/or copyright information of VoiceGenie Technologies Inc. and may not be copied, used, or disclosed without the permission of VoiceGenie Technologies Inc.

© COPYRIGHT 2005 VoiceGenie Technologies Inc.

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, this information is subject to change without notice.

Please note that VoiceGenie makes no warranties with respect to future releases. The information provided herein is for informational purposes only. VoiceGenie reserves the right to change product release schedules and/or features allocated to a product release at any time.

Trademarks

All trademarks are the property of their respective owners. Where those designations appear in this document, and VoiceGenie is aware of a trademark claim, the designations have been printed in initial caps or all caps.



Table of Contents

Proprietary / Copyright Information	3
Notice	3
Trademarks	3
Table of Contents	4
Revision History	5
1 Introduction	6
1.1 Functional Overview	6
1.2 VoiceGenie Components	6
1.2.1 <i>Media Platform (MP)</i>	7
1.2.2 <i>Speech Resource Management (SRM)</i>	7
2 Installation	7
3 System Configuration Overview	7
4 Media Platform Provisioning	9
4.1 Application Development	9
4.1.1 <i>Grammar support</i>	9
4.1.2 <i>Application Example</i>	10
4.1.3 <i>ASR Result Priority</i>	11
4.1.4 <i>TDD specific application properties</i>	12
4.1.5 <i>Limitations</i>	12

Revision History

Version	Date	Change Summary	Author/Editor
1.0	March 30, 2005	Initial release	Johnson Tse
1.1	September 12, 2005	Revised Version	Rob Marchand Lin Chen Andrew Ho

1 Introduction

Telecommunications Devices for the Deaf (TDD) devices, also known as TDD/TTY devices, enable callers to communicate over telephone lines using keyboard devices for input and output.

TDD/TTY operates entirely over the standard PSTN. Unlike a modem, TDD/TTY devices do not negotiate connectivity at the start of the call, nor do they maintain synchronization throughout a call. Rather, they generate the required tones when sending data, and do not transmit at other times.

VoiceGenie 7 has the capability to recognize TDD tones as part of a VoiceXML session. The ability is exposed as part of VoiceGenie's support for Automatic Speech Recognition (ASR).

1.1 Functional Overview

The VoiceGenie TDD/TTY recognition module will work with any TDD/TTY devices that support standard 5-bit baudot encoding with FSK operating at 1800Hz and 1400Hz.

Since 5-bit baudot character sets are case-insensitive, the TDD recognition engine assumes the characters accepted are all capitalized.

A typical call with the VoiceGenie platform is capable of operating in one of three modes:

- TDD/TTY disabled. This mode essentially reflects the current VoiceGenie solution; it does not detect or respond to TDD/TTY input, nor does it render any output in a TDD/TTY compatible format. Any requests by the application to collect input using TDD/TTY will be ignored.
- TDD/TTY input enabled with optional switching. In this mode, the VoiceGenie solution operates as it does currently upon receiving a call. However, it allows the application to utilize TDD/TTY to collect input. It also supports optional switching:

After X seconds (where X is a configurable value) of elapsed time, if no TDD/TTY tones have been detected, the call can switch to mode # 1 above, where TDD/TTY input is disabled.

If a TDD/TTY tone has been detected, the call can switch to mode # 3 below, where a TDD/TTY is assumed to be utilized.

- TDD/TTY only. In this mode, speech input using traditional ASR engines will be disabled; DTMF and TDD/TTY input will continue to be enabled. It will also be possible to force all output to occur using TDD/TTY, if desired by the application.

1.2 VoiceGenie Components

Before explaining the VoiceGenie architecture and provide examples of configurations, the following subsections describe the functionality of each VoiceGenie component. For a detailed VoiceGenie 7 system overview, please refer to the [VoiceGenie 7 Media Platform Users' Guide](#).

1.2.1 Media Platform (MP)

The VoiceGenie Media Platform contains a VoiceXML interpreter that allows incoming/outgoing calls to interact with callers using VoiceXML dialogs. The Media Platform connects to Voice over IP (VoIP) networks or, using PSTN media processing boards, to TDM telephony networks.

1.2.2 Speech Resource Management (SRM)

The Speech Resource Management server is a component that keeps track of all Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) engines and distributes requests from the Media Platform to these engines.

2 Installation

As a pre-requisite, the system must have the following installed:

- VoiceGenie 7 Software or later
- VoiceGenie SRM Server
- VoiceGenie TDD/TTY package

Please refer to Section 6 "ASR and TTS Software Installation" of the [VoiceGenie 7 Media Platform Users' Guide](#) for detailed instructions for how to install the TDD/TTY packages.

3 System Configuration Overview

The VoiceGenie TDD/TTY component is implemented as a shared object loaded by the SRMServer. The SRMServer exposes the TTY engine as an ASR resource.

Once the package is deployed successfully, the system will be configured automatically. From the Cluster Management Platform (CMP) System Management Console (SMC), the Speech Resource Manager sub-menu for the SRM Server provides access to the SRMServer configuration, as shown:

<input checked="" type="checkbox"/>	vrmservice.tty.audiobuffersize	540
<input checked="" type="checkbox"/>	vrmservice.tty.enginename	tyrecognizer
<input checked="" type="checkbox"/>	vrmservice.tty.virtual_dir	tyrecognizer
<input checked="" type="checkbox"/>	vrmservice.tty.numsession	36
<input checked="" type="checkbox"/>	vrmservice.tty.numgramthread	5
<input checked="" type="checkbox"/>	vrmservice.tty.numthread	5
<input checked="" type="checkbox"/>	vrmservice.tty.default.noinputtimeout	10
<input checked="" type="checkbox"/>	vrmservice.tty.default.termdigittimeout	10
<input checked="" type="checkbox"/>	vrmservice.tty.default.interdigittimeout	10
<input checked="" type="checkbox"/>	vrmservice.tty.translatespace	
<input checked="" type="checkbox"/>	vrmservice.tty.audioprocesssize	1320
<input checked="" type="checkbox"/>	vrmservice.tty.audiothreshold	8

The following table describes each of the parameters. These can be changed using the VoiceGenie SMC interface:

Parameter Name	Default Value	Description
vrmservice.tty.enginename	tyrecognizer	The engine name to be used in the CMP health monitor
vrmservice.tty.virtual_dir	tyrecognizer	The MRCP URI to be exposed by the SRM server for incoming ASR requests
vrmservice.tty.audiobuffersize	540	Specifies the internal audio buffer size in bytes. A smaller value results in higher accuracy and rapid response but requires more CPU resources.
vrmservice.tty.numsession	36	The maximum number of concurrent sessions that the TTY Engine will accept.
vrmservice.tty.numgramthread	5	Number of TTY Grammar Threads for the engine. This parameter depends on system performance
vrmservice.tty.numthread	5	VRMServer.TTY.numThread
vrmservice.tty.default.noinputtimeout	10	The default 'no input' timeout value in seconds
vrmservice.tty.default.termdigittimeout	10	The default 'termdigittimeout' timeout value in seconds
vrmservice.tty.default.interdigittimeout	10	The default 'interdigittimeout' timeout value in seconds

vrmserver.tty.translatespace	<empty>	The character with which to replace a space character. If this is not specified, any space character encountered will be dropped in the recognition result. This has been added to support ease of grammar matching for words like "Thank You" and "Thank You".
vrmserver.tty.audioprocesssize	1320	This is the "Jitter" buffer size (in bytes) whereby the audio stream is processed in the specified fixed size at a time.
vrmserver.tty.audiothreshold	8	The audio threshold level for echo cancellation. The recommended value is 8. Specifying a number less than 0 disables this feature. Note: This is an arbitrary level and doesn't correspond to decibels (dB).

4 Media Platform Provisioning

After the TDD/TTY package has been installed, the MediaPlatform -> SpeechResources submenu will show the TDD/TTY engine as follows:

Speech Resource Name:

Hostname/IP: Port:

Resource URI:

Resource Type:

Parameter Name: Value:

Parameter Name: Value:

Parameter Name: Value:

At this point, the user should click on "Target as Primary" button and select the server that has the Media Platform installed. Click on update.

Please note that since the TDD/TTY engine does not have an HTTP fetching capability - therefore the MP speech resource parameter "vrms.client.SendGrammarContent" must be set to true. This setup is performed automatically by the installation package already. Further to this, only in-line grammars are currently supported.

4.1 Application Development

The following sections describe the steps in building a TDD/TTY application:

4.1.1 Grammar support

The TDD/TTY engine accepts W3C GRXML as the format for defining a grammar for the recognition. An example is given below:

```
<?xml version="1.0"?>
<grammar mode="tdd" version="1.0" root="Help-dtmf"
  type="application/grammar-xml"
  xmlns="http://www.w3.org/2001/06/grammar">
  <rule id="Help-dtmf" scope="public">
    <item repeat="1-200">
      <one-of>
        <item>DICTIONARY</item>
        <item>PAPER</item>
        <item>ENGINEERING</item>
        <item>MSG</item>
      </one-of>
    </item>
  </rule>
</grammar>
```

In the example grammar above, the TTD/TTY engine will recognize any of the words PAPER, DICTIONARY, ENGINEERING, MSG, as typed by the user. Notice that the "mode" attribute must be set to "tdd" for TDD enabled grammars. If the same grammar is used for both voice and tdd recognition, then the user can specify speech as well as tdd, for example, by setting mode=" voice tdd".

4.1.2 Application Example

An application that performs TDD recognition will be very similar to a conventional ASR application. An example is given as follows:

```
<?xml version="1.0"?>
<vxml version="1.0">
<meta name="application" content="TDD test"/>
<property name="LOGLEVEL" value="2"/>
<property name="bargein" value="true"/>
<property name="asengine" value="TTY"/>
<property name="confidencelevel" value="0.45"/>
<property name="universals" value="none"/>
  <var name="counter" expr="0"/>
  <form>

    <field name="pword">
      <prompt> <audio src="tty_4wordprompt.vox"/> </prompt>
      <grammar mode="tdd" version="1.0" root="Help-dtmf"
        type="application/grammar-xml"
        xmlns="http://www.w3.org/2001/06/grammar">
        <rule id="Help-dtmf" scope="public">
          <item repeat="1-200">
            <one-of>
              <item>9</item>
              <item>8</item>
              <item>7</item>
              <item>6</item>
              <item>5</item>
              <item>4</item>
              <item>3</item>
              <item>2</item>
              <item>1</item>
              <item>0</item>
```

```

                </one-of>
            </item>
        </rule>
    </grammar>
    <catch event="noinput">
        I did not hear you
    </catch>
    <catch event="nomatch">
        I did not understand
    </catch>
    <filled>
        you said <value expr="pword"/>

        <assign name="counter" expr="counter+1"/>

        <if cond="counter==4">
            <disconnect/>
        <else/>
            <clear namelist="pword"/>
            <reprompt/>
        </if>

    </filled>
</field>
</form>
</vxml>

```

In the above example, the user can input a digit string containing any of the digits from 0 to 9 – the length of the digit string can vary from 1 to 200 digits.

For TDD devices that support dual voice/TDD input modes, the application must play a TDD prompt to “trigger” the device to operate in TDD mode. In the example given, the “tty_4wordprompt.vox” is an audio file (ulaw format) that contains the TDD Prompt.

4.1.3 ASR Result Priority

The priority for the TDD and ASR results can be configured through the SMC, using the “Configuration -> Media Platform” submenu. By default, they have the following values:

```
speech_asrengine_priority = 1
tdd_engine_priority = 0
```

The priority scheme is useful when simultaneous Speech / TDD recognition sessions occur. It determines whether to wait for ASR result before stopping the TDD recognition engine.

For ASR engines with priority 0, when a “start-of-speech” event comes from the engine, it will stop recognition on all the other engines, including other engines with priority 0. For ASR engines with priority n greater than 0, when a “start-of-speech” event comes from the engine, it will stop recognition on all the other engines with priority greater than n , *but recognition will continue on engines with priority equal to n .*

By setting the `tdd_engine_priority` to be higher than the normal speech recognition engine, it allows the TDD recognition to continue when a start-of-speech is detected from the normal ASR engine. This is

useful because if the ASR Engine is using energy-based start-of-speech detection, it will generate a start-of-speech even when the user inputs TDD tones.

In normal scenarios, this priority setting does not need to be changed.

4.1.4 TDD specific application properties

To enable the TDD specific application properties, the following properties must be set in the VoiceXML application:

```
<property name="com.voicegenie.TDDPARAMINUSE" value="true"/>
```

The following properties are supported:

Parameter name	Description
com.voicegenie.tddtermchar	A termchar must be non-empty when specified, and is entered by the user before the tddinterdigittimeout expires, to signify that the users input is complete; the termchar is not included as part of the recognized value
com.voicegenie.tddtermdigittimeout	<p>If the entry of the last character input has brought the grammar to a termination point at which no additional character is allowed by the grammar, then the tddtermdigittimeout may be used in the following scenario:</p> <p>If the tddtermdigittimeout is 0s (the default), then the recognized value is returned immediately after the last DTMF allowed by the grammar, without waiting for the optional termchar.</p> <p>If tddtermdigittimeout is not 0 and if the termchar is non-empty, then the user can enter an optional termchar DTMF. If the user fails to enter this optional DTMF within termtimeout, the recognition ends and the recognized value is returned</p>
com.voicegenie.tddinterdigittimeout	The interdigittimeout determines when the nomatch event is thrown because a TDD grammar is not yet recognized, and the user has failed to enter additional input

4.1.5 Limitations

Following are the set of known limitations for the TDD/TTY implementation:

- Grammars must be defined inline with the MRCP DEFINE-GRAMMAR request, as the TTY engine does not have a fetching module
- Only simple DTMF-style XML grammars are supported currently.

- URI references using rules are not supported (i.e., URI in the form of http://<grammar name>#<rulename>)
- External rule references inside the grammar are not supported
- The grammar matcher class does not perform DTD validation on the grammar
- The TDD/TTY engine does not support lexicon definitions or references;
- The TDD/TTY engine supports only characters that can be defined in the standard baudot encoding
- The VoiceGenie Platform supports both per-utterance and per-call ASR sessions. The TDD/TTY engines supports only the per-call mode. TTY phones do not send FIGS & LTRS mode information with each character, so mode information will be lost in a per-utterance scenario. For FIGS mode (0-9 and space only), both per-call and per-utterance modes are supported.
- Sometimes when the user barges in, the TDD device will still display the remainder of the prompt. This is due to buffering of received TDD audio prior to display
- The recognition performance is known to degrade as the TDD input length increases, especially if the user enters a long series of TDD inputs.