VXI*

The VoiceXML Browser for Asterisk®

INSTALLATION GUIDE

Version: 8.0

i6net*

VXI* – VoiceXML Browser

INSTALLATION GUIDE

Company Name:

Address/City/State/Zip:

Country:

About I6NET

I6NET Solutions and Technologies Limited is a pan-European company specialized in the development of new applications and advanced communication solutions. I6NET creates new business solutions and opportunities with voice interactivity, helping phone and data networks convergence. Its innovative voice browsers systems and software components enable the creation of voice & video services in VoiceXML. You can contact us by email or call us by phone and leave us a message here.

Main/Sales office quarters:
C/ Magallanes 13 – 5º Izq 28015 Madrid (Spain)
VAT Number ES-B83388306 - See more information: www.i6net.com

Table of Contents

1 System Requirements

This manual covers the installation and configuration of the full VoiceXML Browser to using its various advanced features.

Hardware requirements

Recommended configuration:

- CPU Pentium III 2 GHz (minimum)
- 1 Gb of RAM per core
- Motherboard with 533MHz/800MHz system bus support
- Ethernet Network board (with Mac Address)
- TDM board Digium E1/T1 (TE220 / TE440) or compatible

Make sure that your servers are installed and equipped with all the required hardware.
Please check that other hardware installed on your server (such as network adapter, RAID controller, etc.) is supported by the Linux distribution that you use.

Operating System requirements

The VXI* VoiceXML browser runs on most modern Linux Operating Systems (Debian, Redhat, Mandrake, Suse, Fedora, etc). However, it may be difficult for a novice system administrator to install and configure the operating system to meet all of the requirements and provide the best performance. Installing all the software could also prove to be a time-consuming task.

We provide a seamless way to perform a complete server installation from scratch, in less than 15 minutes. The installation uses tar.gz packages and bash scripts.

VXI* has been tested on the following Linux distributions:

- Debian - Debian 4 / 5 / 6 (recommended dev. environment) | http://www.debian.org
- Ubuntu - Ubuntu Server | http://www.ubuntu.com
- Redhat - Redhat Enterprise Server 4 / 5 | http://www.redhat.com
- Fedora - Fedora Core 3 | http://fedoraproject.org
- Suse - Open Suse | http://www.opensuse.org
- Mandrake - Mandrake Linux | http://www.mandriva.com
- AsteriskNOW - Digium Asterisk | http://www.asterisknow.org

NOTE:
Disable the SELinux. It can lock the library loading.
You can use '/usr/sbin/setenforce 0' before launching the VXI* VoiceXML browser.

I6NET uses Debian distribution to generate all packages provided.
Most packages built are specifically configured by default for this Linux OS.

There are two versions of the package that are available:

- "32bit i686" package, compiled with GCC 4.x
- "64bit i686" package, compiled with GCC 4.x

File descriptors:

On a busy Linux system, you may need to increase the number of file descriptors available.

System-wide descriptors are set via /proc/sys/fs/file-max

The per-user limit is still low at 1024
As root, you can issue the command (e.g. in safe_asterisk):

```
ulimit -n 65535
```

However, if running Asterisk as non-root, then edit via:

```
/etc/security/limits.conf
```

```
asterisk soft nofile 65535
asterisk hard nofile 65535
```

To find out how many file descriptors are being used (lsof is a single package command for debian):

```
lsof -p ´pidof asterisk´
```

2 Installation

2.1 Install Asterisk

Installation from Sources

The Asterisk source code can be obtained either through FTP/HTTP or SVN.

The VXI* VoiceXML browser was tested with an Asterisk 1.4/1.6/1,8 standard open souce code go to:

http://www.asterisk.org/downloads

The file downloaded will need to be extracted before compiling. Use the GNU tar application to extract the source code from compressed archive.

This is a simple process that can be achieved through the use of the following commands:

```
# cd /root/src
# tar zxvf asterisk-*.tar.gz
```

Subversion is the best way to keep on the bleeding edge of source releases. If you are wanting to help develop for the Asterisk project, you will want to use SVN to get the most up-to-date source code.
Commands to check out code from our SVN repository:

```
# cd /root/src
```

```
# svn checkout http://svn.digium.com/svn/asterisk/trunk asterisk
```

Commands to get the current snapshot from the release branch of SVN:

```
# svn checkout http://svn.digium.com/svn/asterisk/branches/1.2 asterisk-1.2
```

You can check out the source at any level of the file system. This includes something like svn checkout http://svn.digium.com/svn/asterisk. However, it would be a bad idea to do so, because you will end up checking out the code for every branch and tag that exists in the asterisk repository. Make sure you are careful when checking out the code!

Asterisk is compiled with gcc through the use of GNU make program. To get started compiling Asterisk, simply run the following commands (replace version with your version of Asterisk).

```
# cd /root/src/asterisk-version
# make clean
# make
# make install
# make samples
```

Run the “make samples” command to install the default configurations files. Installing these files (instead of configuring each manually) will allow you to get your Asterisk system up and running much faster.
If your are using a system that makes use of /etc/init.d directories, you may wish to run the “make config” command as well.

This will install the startup scripts and configure the system to execute Asterisk automatically at startup.

Installation from Packages (recommended)

I6NET provides a free and compiled Asterisk installation package. The install package contains a minimal amount of default configuration files to get started, and provides an efficient way to get your Asterisk system up and running.

First, unzip/untar the Asterisk package by using the command:

```
# tar xvzf asterisk_Vx.x.x_date.tar.gz
```

Next, go to the directory of the Asterisk package generated and type the following command:

```
host:~# cd asterisk_Vx.x.x_date
host:~/asterisk_Vx.x.x_date# ./install.sh
--- Asterisk IP/PABX Vx-x-x Installation ---
Creating directories...
Installing asterisk binary...
Installing configuration files...
Installing sounds...
Installing modules...
--- Asterisk IP/PABX Vx-x-x installation has finished ---
host:~/asterisk_Vx.x.x_date#
```

NOTE:
If your Asterisk is already installed or you are installing from sources please check your are using:

Asterisk 1.2	=>	VXI* 1.4 package (discontinued)
Asterisk 1.4	=>	VXI* 3.X, 4.X, 5.X, 6.X package
Asterisk 1.6	=>	VXI* 3.X, 4.X, 5.X, 6.X package
Asterisk 1.8	=>	VXI* 6.X package (stable)

2.2 Install Dahdi (optional, only for TDM)

The last Asterisk releases now support Dadhi driver.
If your system use a TDM card, you must install first Dadhi drivers to manage T1/E1 interface. ü The Dadhi drivers and tools should be compiled in the server. You need to install a building environment (compiler, binutils and kernel headers). For the Debian distribution, install the packages : linux-headers-`uname -r`, make, gcc.

Example:

```
apt-get install linux-headers-`uname -r`
```

(Packages to be able to compile : binutils, make, gcc)

This Dahdi packages associated to the i6net Asterisk package or in the Asterisk installation directory.
You can download the latest Dadhi sources files (from www.asterisk.org) (take care with the compatibility with the Asterisk binaries build by i6net):

- dahdi-linux-x.x.x.x.tar.gz
- dahdi-tools-x.x.x.x.tar.gz

Install the Dahdi driver:

```
# tar xvfz dahdi-linux-x.x.x.x.tar.gz
# cd  dahdi-linux-x.x.x.x
# make
```

```
# make install
```

Results:

```
[…]
###################################################
###
### DAHDI installed successfully.
### If you have not done so before, install the package
### dahdi-tools.
###
###################################################
```

Install the Dahdi tools:

```
# tar xvfz dahdi-tools-x.x.x.x.tar.gz
# cd  dahdi-tools-x.x.x.x
# ./configure
# make
# make install
# make config
```

Results:

```
[…]
DAHDI has been configured.
```

If you have any Dadhi hardware it is now recommended you edit /etc/dahdi/modules in order to load support for only the Dadhi hardware installed in this system. By default, support for all Dadhi hardware is loaded at Dadhi start.

```
I think that the DAHDI hardware you have on your system is:
pci:0000:0b:08.0     wct4xxp-     d161:0220 Wildcard TE220 (4th Gen)
```

Configure the Dahdi driver:

Configuration files are not stored in /etc/dahdi:

```
# cd /etc/dahdi
# ls
init.conf  modules  system.conf
```

Example of system.conf (dual E1 board):

```
#
# Dahdi Configuration File
#

span=1,1,0,ccs,hdb3,crc4
bchan=1-15
dchan=16
bchan=17-31

span=2,1,0,ccs,hdb3,crc4
bchan=32-46
dchan=47
bchan=48-62

#span=3,1,0,ccs,hdb3,crc4
#bchan=63-77
#dchan=78
#bchan=79-93
```

```
#span=4,1,0,ccs,hdb3,crc4
#bchan=94-108
#dchan=109
#bchan=110-124

loadzone=es
defaultzone=es
```

You can disable the unused modules by editing the /etc/dahdi/modules and removing or commenting them. Example of modules (dual E1/T1 board wct4xxp):

```
#
# Dahdi modules
#

wct4xxp
```

Start / Stop Dahdi driver:

The Dahdi tools install a startup script, /etc/init.d/dahdi. You may also use this script to control Dahdi from the Linux command line:

```
# /etc/init.d/dahdi start
# /etc/init.d/dahdi restart
# /etc/init.d/dahdi stop
```

NOTE:

Remember that, the Dahdi module loading is disabled in the I6NET packaged Asterisk version. Disable the noload in the /etc/asterisk/modules.conf.

2.3 Install Video IP/3G/RTMP (optional, only for Video)

To use 3G-324m video features, your system must have a TDM card. You don't need install this package, if you are going to use your system only for voice services.

```
# tar xvzf video_VX-X_date.tar.gz
```

```
# cd video_VX-X_date
# ./install.sh
```

2.4 Install VXI* VoiceXML browser

Use root to install VXI*.
Unzip and untar the openvxi package by using the command:

```
# tar xvzf vxml_VX.X_date.tar.gz
```

Go to the directory of the openvxi and type the following command.

```
# cd vxml_VX.X_date
# ./install.sh
```

NOTE:
VXI* is now using new libraries and compiler version, before installing the VoiceXML browser please check your Operating Systems has the a GCC3 o GCC4 libraries environment :

VXI* 1.4 is build with GCC3.3 / libstdc++5
VXI* 1.5, 2.X, 3.X, 4.X, 5.X, 6.X are build with GCC4 / libstdc++6

3 Setup

3.1 Check Asterisk setup

Start Asterisk on your server using the command:

```
# asterisk –cvvvvv
```

The number of v's doesn't really matter, the more there are the more verbose asterisk will be in its display, but don't omit the c from the end. When you do this you will see a great deal of information scroll past quite quickly, don't worry about this, just wait until you get a prompt like:

```
*CLI>
```

To see the installed version:

```
*CLI> show version
```

First testing call (echo test), using a SIP phone:

```
SIP:600@<your server IP address>
```

3.2 Launch VXI* deamon

The VoiceXML browser software is installed in /usr/sbin and /usr/lib/openvxi. The VoiceXML browser setup script on Linux is /etc/init.d/openvxi. The openvxi script calls the /usr/sbin/safe_openvxi executable that functions as a monitor and auto-loader for your VoiceXML browser system. This safe_openvxi starts VoiceXML browser and monitors it to make sure it is still running. If the VoiceXML browser process dies, the script will attempt to restart it.

```
# /etc/init.d/openvxi start
```

To stop the VXI* deamon:

```
# /etc/init.d/openvxi stop
```

NOTE:
This startup script runs only for Debian/Ubuntu Linux distributions, please modify or install a correct this script file to launch correctly VXI* from other Linux systems. We provide a script to start the actual, AsteriskNOW linux distribution from DIGIUM.

3.3 Launch Asterisk deamon

Our packages install a startup script, /etc/init.d/asterisk.
You may also use this script to control Asterisk from the Linux command line:

```
# /etc/init.d/asterisk start
# /etc/init.d/asterisk restart
# /etc/init.d/asterisk stop
# /etc/init.d/asterisk status
```

NOTE:
This startup script runs only for Debian/Ubuntu Linux distributions please modify or install a correct file to launch correctly VXI* from other systems. We provide a script to start the actual, AsteriskNOW linux

distribution from DIGIUM.

3.4 Check VXI* setup

At the end, if everything is OK you will have one more application added – Vxml.
You can check this by starting (or restarting) the Asterisk and using the show applications command.

```
# asterisk –r
```

```
*CLI> help vxml
```

To show the license status (DEMO) use only one channel:

```
*CLI> vxml show license
```

4 Management

4.1 Managing VXI*

Once you have completed installing the VoiceXML browser, you must configure the VoiceXML browser before attempting to place calls. The management and configuration procedures completely depend on the integration, as the reference VoiceXML browser components never directly access a configuration subsystem. Instead, all configuration parameters are passed to the initialization or resource creation functions for each VoiceXML browser component.

OpenVXI is an old portable open source VoiceXML interpreter and is a part of VXI*. The VoiceXML browser is configured through a text configuration file specified on the command line. Each line of this file defines a value for a property (parameter) applying to a given OpenVXI component; the possible property names are defined in the interface header files for the various components. The configuration file format is described in detail in a comment block at the top of the sample configuration file, /etc/openvxi/client.cfg.

NOTE:

Before any configuration update, make a backup copy of your client.cfg file.

Once you have made a backup copy, review the properties in the configuration file and modify their settings as appropriate. For an explanation of each property, refer to support. In most cases you can modify the following properties to get started (listed in priority order):

- Configure the internet access to use http
- Configure the cache management
- Activate/ deactivate traces
- Configure the supported MIME types

4.2 Cache Management

In the Client configuration section, set client.inet.cacheDir to the directory where you want to store cached Internet files from URL fetches.

Set the client.inet.cacheTotalSizeMB property to the desired size limit in megabytes for the Internet fetch cache.

Processing VXI* produces standard traces under the /tmp/log.txt file. The trace file names are referenced in the /etc/openvxi/client.cfg file. Their format can be found in the Log Files section of this document.

NOTE:

To disable the cache entry, set the maxage and maxstale properties to 0. This properties can be change in the default.xml file to globally disable the cache.

4.3 Webserver Configuration (optional)

The VXI* VoiceXML browser supports running against static content stored as local files as well as static or dynamic content served by a web server. The VoiceXML browser is generally used with a web server.

Content can be VoiceXML documents, audio / images / video files (if supported by your version or license), and grammar files. When a web server delivers the content, it also provides the MIME content type for the content and information about the desired caching behavior for the content.

For the MIME content type for VoiceXML documents, a value of application/vxml+xml is recommended, but not mandatory, as the VoiceXML browser can use information within the XML document to verify it is a VoiceXML document. For prompt, recognize and record interfaces the MIME content type is more important. Support for the following audio formats is recommended for play and record operations, but not mandatory.

MIME Content Type	Description
audio/basic	RAW (headerless) 8kHz 8-bits mono mu-law [PCM] single channel. (G.711)
audio/x-alaw-basic	RAW (headerless) 8kHz 8-bits mono A-law [PCM] single channel. (G.711)
audio/x-wav	WAV (RIFF header) 8kHz 16-bits mono [PCM] single channel.
audio/X-WAV	WAV (RIFF header) codec GSM (6.10) 7-bits mono single channel.
audio/x-gsm	GSM (6.10) 7-bits mono single channel.
video/mp4	MP4 (Iso file) H.263 7 fps, mulaw/alaw/AMR-NB
video/3gp	3GP (Iso file) H.263 7fps, AMR-NB

The length of time the VoiceXML browser caches an item fetched from a web server is controlled by the caching values returned in the HTTP/1.1 header. The VoiceXML browser currently only supports the “Expires” header, which defines the date and time when the VoiceXML browser must discard the cached copy and re-fetch from the web server. Some web servers allow defining caching property defaults based on the MIME content type, useful for allowing caching of audio files but not dynamically generated VoiceXML documents, for example.

*4.4 Start / Stop VXI**

The VoiceXML browser software is installed in /usr/sbin and /usr/lib/openvxi. The VoiceXML browser setup script on Linux is /etc/init.d/openvxi. The openvxi script calls the /usr/sbin/safe_openvxi executable that functions as a monitor and auto-loader for your VoiceXML browser system. This safe_openvxi starts the VoiceXML browser and monitors it to make sure it is still running. If the VoiceXML browser process dies, the script will attempt to restart it.

To load or unload the VoiceXML browser, use the “openvxi start” and “openvxi stop” commands:

```
#/etc/init.d/openvxi start
#/etc/init.d/openvxi stop
```

Starting processes at boot time are different among the different operating systems so it is best to consult your OS documentation on how to do this.

In Debian-based systems you can add this to your /etc/init.d file, however this will not shutdown the VoiceXML browser very cleanly during a reboot or shutdown.

In a Debian environment you may be able to get a working /etc/init.d script by running the following commands. It will then run the following command:

```
#/sbin/chkconfig --add openvxi
```

4.5 Using VXI*

The VoiceXML browser installs the app_vxml Asterisk application that uses the browser to execute VoiceXML pages.

Check the VXML application module

For more information on applications, just type "show application" at the Asterisk CLI prompt.

To show details of how you use that particular application in this file (the Asterisk Dial plan).

Type:

```
*CLI> show application <command>
```

For example, the vxml application:

```
*CLI> show application vxml
```

4.6 Usage Syntax

Description:

Launch a VoiceXML session in the channel and when complete, return control.

The URL can by set with different ways :

- Pass the URL as the appplication parameter. Exemple : vxml(file:///tmp/test.vxml)
- Pass the account name or the account number. Exemple: vxml(test)
- Pass the “@” to match the caller number to the account number.
- Set the VXML_URL variable before executing the vxml application.
- Don't pass any parameter, and use the configuration accounts section. If account number match with the called number, its URL(s) and parameters will be use.

The following describes how to execute a VoiceXML session.

Syntax:

```
Vxml([URL|Name|Number])
```

Variables:

VXML_URL

If the variable VXML_URL has been set when vxml runs, the value of that variable will be used for the URL unless the parameter is not set to the application.

VXML_ID

If the variable VXML_ID has been set when vxml runs, the VoiceXML session ID variable called “telephone.id” is set with this value (in the VoiceXML execution session context).

VXML_PARAM

If the variable VXML_PARAM (VXML_AAI is an alias) has been set when vxml runs, the value of that variable will be used as "telephone.param" (in the VoiceXML execution session context) and session.connection.aai.

VXML_RESULT

After execution, the VoiceXML result of <exit> tag and the property 'expr' are accessible by the variable VXML_RESULT.

Example:

```
[incoming]
exten => s,1,Answer
exten => s,n,Wait(3)
exten => s,n,Vxml(http://vxml.i6net.com/index.vxml)
exten => s,n,Hangup
```

4.7 Asterisk Online Help

Online help can be accessed by typing the following command at the CLI prompt:

```
*CLI> help vxml
```

4.8 Management Commands

Now your VXI* VoiceXML browser and Asterisk PBX are running, you can manage the VXML service using the Asterisk prompt *CLI>:

`*CLI> vxml debug`
Enable VoiceXML debugging for the Asterisk application.

`*CLI> vxml debug interpreter all`
Enable VoiceXML debugging for the interpreter application ().

`*CLI> vxml no debug`
This command disables VoiceXML debugging for the Asterisk application.

`*CLI> vxml no interpreter debug`
This command disables VoiceXML debugging for the interpreter application.

`*CLI> vxml show cache / vxml cache show`
Print the files in cache (if debug mode is enabled) and/or the number of file in the cache.

`*CLI> vxml cache clear`
Delete all the files in the cache.

`*CLI> vxml cache purge`
Delete all the files in the cache older than the maxage parameter set in the configuration.

`*CLI> vxml show license`
Use this command to show the license information.

```
*CLI> vxml reload
```
Reload the configuration.

```
*CLI> vxml show configuration
```
Use this command to show the configuration summary of VoiceXML interpreter.

```
*CLI> vxml show accounts
```
Show the accounts configured.

```
*CLI> vxml show account <number>
```
Show the accounts details of the account ID specified.

```
*CLI> vxml show statitiscs
```
Provides a dump statistics on VoiceXML interpreter

```
*CLI> vxml show dates
```
Provides a dump dates on VoiceXML interpreter

```
*CLI> vxml show sessions
```
Provides a dump sessions on VoiceXML interpreter.

```
*CLI> vxml show session
```
Provides a full dump session on VoiceXML interpreter.

```
*CLI> vxml originate chantype/number=application(parameters)
```
Originate an outgoing call, you can request to use a VoiceXML session.

The following entries are the Asterisk CLI commands for the VoiceXML browser.

Example:

Add extensions to the Asterisk dial plan /etc/asterisk/extensions.conf:

```
exten => 888,1,Answer
exten => 888,n,Wait(3)
exten => 888,n,Vxml(file:///root/example.vxml)
exten => 888,n,Hangup
```

You can create and edit the file /root/example.vxml with the GNU text editor, VI, for example.

NOTE:

This example will work if you have text-to-speech configured. If not, use a pre-recorded wav or gsm file to replace the “Hello world!” text by an <audio> tag. For more information, see the format extensions supported by Asterisk.

```
<block><audio src=”hello.wav”/></block>
```

Save the file in the same directory as the VoiceXML script (relative reference in this example).

Reload the extensions configuration with:

```
*CLI> extensions reload
```

Call the service by calling:

```
SIP:888@<your server address>
```

4.9 Troubleshooting (for Technical Support)

This chapter covers troubleshooting procedures for the VoiceXML Browser, describing some basic techniques that can be used when working with VXI*.

Collecting Information for Technical Support

As part of the process of reporting problems, download log files and core files from the VoiceXML Browser and send them to I6NET, together with the current configuration files.

- /tmp/log.txt (default configuration)
- All core files

Collected files should be sent by email to Technical Support (support@i6net.com).

Log Files

The log files contain information about the operation of the VoiceXML Browser. The file is /tmp/log.txt, which details the VoiceXML processing on the VoiceXML Browser.

If a failure occurs and you need to contact I6NET support (at support@i6net.com), they may ask you to activate traces to allow analysis of the system functions and make a complete appraisal of the problem. To do so you must follow these procedures:

Edit the configuration file client.cfg in /etc/openvxi/.

The levels are defined by these lines which are the# API/general log traces for each component:

```
client.log.diagTag.2000                     VXIInteger  0
client.log.diagTag.2001                     VXIInteger  0
client.log.diagTag.3000                     VXIInteger  0
client.log.diagTag.3001                     VXIInteger  0
client.log.diagTag.3002                     VXIInteger  0
```

The ranges are associated to different interfaces:

```
200x:  Cache interface
300x:  Internet HTTP interface
400x:  ECMAscript interface
500x:  Prompt interface
600x:  Recognize interface
700x:  Telephony/Session interface
800x:  XML Interpreter
900x:  Object interface
1000x: Main/Client application
```

To enable a level, set the 1 value and 0 to disable.

To validate the modifications, the VoiceXML interpreter must be restarted. A simple way to do this is to stop and restart this application with the Asterisk and OpenVXI script:

```
# /etc/init.d/asterisk stop
# /etc/init.d/openvxi stop
# /etc/init.d/openvxi start
# /etc/init.d/asterisk start
```

The log file is generated in the temporary directory, and is named log.txt. The location and filename are configurable. To purge the file type:

```
# > /tmp/log.txt
```

NOTE 1:

Never delete the /tmp/log.txt directly, otherwise you should restart the VoiceXML browser to generate a new one.

NOTE 2:

An Apache/PHP script exists generate the traces from a standard Internet browser (Internet Explorer, Mozilla/firefox...). Get it from our web site

NOTE 3:

At the end of the trace record, don’t forget to stop it to recover optimal real time function. With the V4.x release, you can dynamically enable/disable the interpreter traces with an Asterisk *CLI> command.

Simple traces for the VoiceXML development :

```
*CLI> vxml debug interpreter dev
```

Full traces :

```
*CLI> vxml debug interpreter all
```

Disable the interpreter traces :

```
*CLI> vxml no debug interpreter
```

Restore the log traces :

```
*CLI> vxml debug interpreter log
```

Current Sessions

To determine how many VoiceXML sessions are currently active on the system, use the statistics dump on the CLI. This command displays the current number of sessions on the VoiceXML Browser:

```
*CLI> vxml show statistics
```

File descriptors

To find out how many file descriptors are being used follow this:

Find out program PID:

```
# ps -ef | grep asterisk
```

Find out program PID (other command):

```
# pidof asterisk
```

List of files opened by PID (details):

```
# ls -l /proc/[PID]/fd
```

List of files opened by PID (counter):

```
# ls -l /proc/[PID]/fd | wc -l
```

More information about file descriptors:

```
# lsof | grep "[PID]"
```

5 Configuration

5.1 Configuration file: vxml.conf

The configuration file is /etc/asterisk/vxml.conf. It is editable by typing:

```
# vi /etc/asterisk/vxml.conf
```

Sections format:

```
; commented text starts with a “;”

[general]
...

[control]
...

[license]
...

[account1]
...

[account2]
...

[accountn]
...
```

The vxml.conf file is divided in serveral sections to set different group of parameters and functions:

- general
- control
- license
- accounts

NOTE:

Please note that before configuring the Asterisk Module, you should make a backup copy of vxml.conf. VXI* upgrade process keep the current configuration but some specific updates can be mandatory for future releases.

5.2 General Section (general)

The following section covers general functions of VXI*.

debug= {0/1}

The default is 0, meaning no debug. If the function is set with a value greater than 0, the VoiceXML Asterisk application will generate debug traces.

autoanswer= {yes/no}

The default is yes. If set, the VoiceXML application will answer the Asterisk channel before starting the VoiceXML session.

autohangup= {yes/no}

The default is no. If set, the VoiceXML application will hangup the Asterisk channel at the end of the the VoiceXML session.

autoexit= {yes/no}

The default is yes. If set, the VoiceXML application will force the exit (with -1 return value) of the Asterisk if the connection with OpenVXI is lost.

defaulttimeout= {1..60}

You can set the default timeout value to allow for DTMF recognition, if not set from the VoiceXML syntax. In milliseconds (ms).

The default value is 5s

defaultinterdigittimeout= {1}

The default interdigit timeout value can be set to allow for DTMF recognition, if not set from the VoiceXML syntax.

wavcode= {pcm/gsm}

This function allows you to configure the default codec for the record associated to the MIME type “audio/x-wav”. You can select gsm or pcm. While the default codec is PCM, the audio files recorded could be too big to be easily posted to an HTTP server.

audiosilence= {filename}

The VoiceXML browser can loop and generate a sound during interaction phases. Enter the specified sound file in pcm format (you will need to omit the filename extension). Sound files are stored in the /var/lib/asterisk/sounds directory by default, however the directory path can be changed in asterisk.conf.

The default value is no filename. This function is disabled in this release.

videosilence= {filename}

The VoiceXML browser can loop and generate a video clip during interaction and audio prompting. Enter the specified sound file in Asterisk Raw h263 format (you will need to omit the filename extension). Sound files are stored in the /var/lib/asterisk/sounds directory, however the directory path can be changed in asterisk.conf.

The default value is no filename. This function is disabled in this release.

recordsilence= {yes/no}

The VoiceXML browser can generate an audio silence (RTP packet generated), during the recording phase. This feature has been added to prevent gateway hang-ups if the server does not generate packets during a configured period. If the video mode is selected, it is preferable to unset this option to generate a video echoing during the recording phase (like a mirror).

The default value is no.

threshold= {1...32767}

This value is the minimum threshold, calculated by averaging all of the samples within a frame, for which a frame is determined to either be silence (below the threshold) or noise (above the threshold). This parameter is use to detect the record final silence.

The defaukt value is 256.

dialformat= {application parameters with %s}

This is a string to specify the interface and the peer that has been chosen for the transfer. The “%s” will be replaced by the string set in the <transfer> dest attribute. Remember to prefix the dest value with “tel:” to generate the transfer function. Other prefixes have been added to match some of the Asterisk functions, such as conference, call an application, etc.

The default value is SIP/%s.

dialformatvideo= {application parameters with %s}

This function is like dialformat, but for a video transaction.

The default value is SIP/%s.

blindapplication= {asterik application name or empty}

This is a string to specify the application used with the blind transfer mode. If the application is empty, the VoiceXML application will generate a Dial command with the transfer parameters after the VoiceXML session.

The default value is 'transfer'.

video = {yes/no}

Enable or disable the video feature (controlled by the license key).

The default value is no.

removeprompts= {yes/no}

The VoiceXML interpreter dynamically generates the files to be prompted in the /tmp directory. After prompting, you can choose to delete or keep the files as they can be overwritten by the new <prompt> sections.

The default value is yes.

cachemaxage={-1 or 0-...}

Indicates that the cache is willing to remove contents whose age are greater than the specified time in seconds (set the value -1, to disable the cache purge).

The default value is -1, disabled.

cachetimeout={-1, 0-...}

Cyclic cache check period in seconds (-1 or 0 to disable).

The default value is -1, disabled.

cachehour={-1 or 0-23}

Specific hour to check the cache and purge the older files (-1 to disable).

The default value is -1, disabled.

autoreloadconfig= {yes/no}

Use this function to automatically update the configuration when the file is modified. The update is based on the configuration file date or you can initiate an update to the file at any time on your own. Be careful not to save incomplete configuration files.

The default value is no.

speech= {yes/no/emulation/automatic}

You can enable or disable the ASR (speech recognition) using the VoiceXML application. “Yes” means that the ASR resource will be allocated during the duration of the VoiceXML session. The “emulation” value is a permissive mode that always returns OK, even if the ASR is not present. The “automatic” value will allocate the speech resource if a creation grammar is asked by the VoiceXML interpreter. The speech resource will be released after all grammars are released.

The default value is no.

speechprovider= {lumenvox/verbio}

You can set which speech recognition provider to allocate to the speech resource.

When the default is empty, use the first option.

speechscore= {0...100}

The speechscore function allows you to set the confidence score of the speech recognition engine (ASR). The default number is 50. When the VoiceXML Asterisk application gets an ASR result, the score is returned with the word or phrase recognized. If the score is low, this limiter value is used to return a no match error/event instead a wrong result.

speechforcedscore={0..100}

Force the score value returned to the VoiceXML interprete.

The default value is 0 (to disable the forcing).

speechdirectory= {directory file}

It is preferable to not set this parameter and let the VoiceXML application to get the default values.

The directory file is where the built-in grammars are stored.

For the Verbio ASR, the default value is "" (use internal Verbio builtins)

For the LumenVox ASR, the default value is "/var/lib/openvxi/grammars/ABNF_%s_%s.gram" (ABNF grammars provided by I6NET).

speechunload= {yes/no}

Idisable the call to the “speechunload” application when the grammars are free.

Some providers don't support to send empty 'GRAMMAR-DEFINE'.

The default value is yes (unload the grammars).

cdrupdate= {yes/no}

If value is set to yes, at the end of the VoiceXML session, the VoiceXML application updates the CDR information. The CDR is information generated for each call in order to get data statistics.

After the update at the end of the VoiceXML session, the CDR will be locked.

The default value is yes.

setapp = {“app_vxml” name}

App references the last Asterisk application executed, therefore the setapp value is set with the name of the last application. “App” is also a field of the CDR data.

setuserfield = {exit returned value}

This value is set from a VoiceXML result, specifically from the <exit>, “expr” attribute. “Userfield” is also a field of the CDR data.

accountcode= {VoiceXML account}

This function can be set with the name of the VoiceXML account, if an account was found and the updatecdr option is enabled.

cdrdial= {yes/no}

If value is set to “yes”, the dial application used for the <transfer> will generate a new CDR.If set to “no”, no CDR will be generated and the current CDR will be kept.

The default value is “yes”.

cdrprompt= {yes/no/all}

If value is set to “yes” and the property cdrprompt is set to “true” in the VoiceXML context, the audio/video prompts will generate a new CDR (duration is the duration of the prompt, DTMF indicate the prompt have been skipped by a DTMF, and HANGUP if the prompt is skipped by a hangup).If set to “no”, no CDR will be generated (default value).

If the value is set to “all”, all the audio/video clips will generate CDRs.

The default value is “no”.The alias promptcdr is supported too (in the vxml.conf and for the VoiceXML property name).

cdrconference= {yes/no/all}

If value is set to “yes” and the property cdrconference is set to “true” in the VoiceXML context, the conference transfers will generate a new CDR (duration is the duration of the conference user's sesion, DTMF indicate the conference have been skipped by a DTMF, and HANGUP if the conference is skipped by a hangup).If set to “no”, no CDR will be generated (default value).

If the value is set to “all”, all the conference sessions will generate CDRs.

The default value is “no”.

The alias conferencecdr is supported too (in the vxml.conf and for the VoiceXML property name).

cdroverwrite= {yes/no}

If value is set to “yes”, the CDR field source and destination are overwritten with the contents of the variables VXML_REMOTE and VXML_LOCAL.

The default value is “no”.

cdrspeech= {yes/no/all}

If value is set to “yes” and the property cdrspeech is set to “true” in the VoiceXML context, the ASR/speech will generate a new CDR (duration is the duration allways 0, NOINPUT, NOMATCH, and MATCH are the Lastdata values, the UserField will set with the best result and its score).If set to “no”, no CDR will be generated (default value).

If the value is set to “all”, all the ASR/speech will generate CDRs.

The default value is “no”.

The alias speechcdr is supported too (in the vxml.conf and for the VoiceXML property name).

priorityevents= {yes/no}

If value is set to “no”, the events (single DTMF) are not checked before the end of the full DTMF input, (if there is another complex grammar enabled, with a maxlength > 1).

The default value is “yes”.

recorddirectory= {directory file}

The directory file is where the records are stored. If you want to store localy the recordings, you can use the attribute dest in the tag <record>. Files with relative filenames where stored here. If you use the option 'mark' (from the account ot the dialplan), the VXI module will create directories using the 'mark' and will store the files in this directories.

5.3 Control Section (control)

The control section describes the control functions that allow you to configure DTMF commands that stream a file with fast forward, pause, reverse, restart, etc. To enable a control during a prompt, you must first set the VoiceXML property name to equal “control, with a value of “yes.”

<property name=”control” value=”yes” />

This works only with GSM and WAV files - in all other cases the property is ignored.

forward= {DTMF(s)/empty}

Fast-forward when this DTMF digit is received.

The default value is #.

reverse= {DTMF(s)/empty}

Rewind when this DTMF digit is received.

The default value is *.

stop= {DTMF(s)/empty}

Stop when this DTMF digit is received.

The default value is 0123456789.

pause= {DTMF(s)/empty}

Pause playback when this DTMF digit is received.

The default value is empty.

restart= {DTMF(s)/empty}

Restart playback when this DTMF digit is received.

The default value is empty.

skipms= {delay in milliseconds}

This is the number of milliseconds to skip when rewinding or fast-forwarding.

The default value is 5000.

5.4 License Section (license)

The following section covers license information and keys of VXI*.

To obtain a commercial valid license key for your copy of VoiceXML browser (for use more than one session), you should run asterisk and get your code by typing the following commands:

```
*CLI> vxml show license
```

Default License section (no key) allow to run any VoiceXML application with one free port.

Example:

```
[license]
max=1
key=
```

To test VXI* you have these options:

- Free Trial, 1 port unlimited time. no key required
- Free Trial, 30 ports 30 min for multi-session testing, set key = evaluation

Example:

```
[license]
max=30
key=evaluation
```

List of License options:

max={0..300}

Maximun number of ports (vxml sessions running at the same time)

video={yes/no}

Video activation for video IVR services

texttospeech={yes/no}

Text-to-speech activation to connect a TTS engine

speech={yes/no}

Speech-recognition activation to connect an ASR engine

externals={yes/no}

External Xtras* modules activation

dialer={yes/no}

Dialer for outbound calls

chanh323={yes/no}

H323 specific module activation

chanrtmp={yes/no}

Flash/RTMP Server Channel module activation

key={...}

License key

5.5 Account Section (account_)

The VXI* VoiceXML browser is designed to manage hosted VoiceXML services and share different applications thru different Inbound / Outbound phone lines. This will allow you to manage customers' accounts for VoiceXML hosted applications and control easily the capacity you need to assign to each one.

To create accounts, you need to add an [accountX] section (where "X" is a number 1-100) in the bottom part of the /etc/asterisk/vxml.conf file. Please find here three added accounts to manage different port capacity per application/URI:

Examples:

```
[account1]
name=voiceportal1
local=0099090
url=http://www.i6net.com/vxml/voiceportal1.vxml
max=25

[account2]
name=helloworld
local=0099090
url=http://www.i6net.com/vxml/helloword/index.vxml
max=10

[account3]
name=localservice
local=0099090
url=http://localhost/vxml/test.vxml
max=51
```

The account properties are defined within sections. Each account has a specific section. The sections are named and numbered from account1 to account99 (maximun 100 accounts). The example below shows the functions in each account section.

```
[account1]
name=example
number=*3
url=http://…/index.vxml
urlvideo=http://…/index.vxml
max=5
speech=no
dialformat=SIP/%s
dialformatvideo=SIP/%s
```

name= {string}

This function indicates the name or designation of the account. This reference can be used to identify the account used by the VXML application, Vxml(name of the account). This allows you to execute a VoiceXML session corresponding to the account values.

number= {called number}

This function allows the identification of an account with the called number, signaling information from the ANI function, CALLERID(). You can use the “*”, the wildcard character, to specify a substring such as *03 or 014612* , that the number function will contain. You can start the number with “_” to use Asterisk Diaplan Patterns too (see : http://www.voip-info.org/wiki/view/Asterisk+Dialplan+Patterns).The function can also be used to identify the account, like the name, Vxml(number of the account). You can start the number with “@” to match the caller number to the account number (use the same syntax as the called number

There is a way to)

url= {voicexml URL}

This function indicates the VoiceXML URL of the account.

urlvideo= {voicexml URL}

This function defines the Video VoiceXML URL of the account. The call is identified as a video call by the set codecs or the function CHANNEL(transfercapability)=VIDEO.

max= {0…120}

This indicates the maximum number of sessions allowed to this account. If there are not enough sessions then the VoiceXML application will generate an error.

dialformat= {application(]/%s[)}

This is similar to the general function, but for the account only. If not set, use the general value.

dialformatvideo= {application(]/%s[)}

This function has the same purpose as the dialformat, only for video sessions.

force= {video/audio}

If set to video, set the Transfercapability to VIDEO (and enable the h324m processing).

If set to audio, the Vxml application execute directly the account URL and bypass the redirection execution (parameter with '@').

speech= {yes/no/emulation/automatic}

This speech function is as the general function, but for the account only. If not set, use the general value.

speechprovider= {lumenvox/verbio}

This speech function is as the general function, but for the account only. If not set, use the general value.

speechscore= {0..100}

This speech function is as the general function, but for the account only. If not set, use the general value.

speechforcedscore= {0..100}

This speech function is as the general function, but for the account only. If not set, use the general value.

mark= {string/@local/@remote/@id/@param}

Set a string mark in the VoiceXML browser traces. The session ID and this string will be added to the channel number column (3rd) in the traces (Example : ...|33|... → ...|33_1_user1|...).

Four redirection exist :

@remote : caller number

@local : called number

@id : VoieXML id parameter value

@param : VoiceXML parameter value

durationlimit= {0…}

This indicates the maximum duration in seconds for the session using this account. The call will be hangup after this maximal duration.

Example:

```
; VoiceXML Configuration
;
[general]
wav_codec=gsm
videosilence=;silence
audiosilence=;silence
debug=4
video=yes

[license]
max=100
key=…

[account1]
name=Test
url=http://host.i6net.com/vxml/links/vxml/index.php
max=1

[account2]
name=Test2
url=http://www.mdfactory.com/mpm/camp/index.vxml
max=3
dialformat=SIP/%s@voztele-out
```

To assign an extension to a VXML account just follow this example, where we are assigning the previous account to three extensions number in your /etc/asterisk/extension.conf asterisk:

```
[default]

exten => 981001001,1,Vxml(voiceportal1)
exten => 981001002,1,Vxml(helloworld)
exten => 981001003,1,Vxml(localservice)
```

NOTE:
When you update your vxml.conf file, remember to refresh configuration making a command “vxml reload” in your CLI*> prompt. If you have added SIP, PRI or new extensions you must launch “sip reload”, “extensions reload”, “dialplan reload” or reload asterisk/openvxi processes. Use the command “vxml show accounts” to dump your accounts.

```
CLI*> extensions reload
CLI*> vxml reload
CLI*> vxml show accounts
```

Example:

Add extensions to the Asterisk dial plan /etc/asterisk/extensions.conf:

```
exten => 888,1,Answer
exten => 888,n,Wait(3)
exten => 888,n,Vxml(file:///root/example.vxml)
exten => 888,n,Hangup
```

You can create and edit the file /root/example.vxml with the GNU text editor, VI, for example.

This example will work if you have text-to-speech configured. If not, use a pre-recorded wav or gsm file to replace the "Hello world!" text by an <audio> tag. For more information, see the format extensions supported by Asterisk.

```
<block><audio src="hello.wav"/></block>
```

Save the file in the same directory as the VoiceXML script (relative reference in this example).
Reload the extensions configuration with:

```
*CLI> extensions reload
```

Call the service by calling:

```
SIP:888@<your server address>
```

5.6 Commercial license activation

The VXI* VoiceXML browser is protected by a key license. The license is host-based: each server has a different license key. If the installation is not activated, you can execute a single VoiceXML session (audio). This is the free use mode. To obtain a commercial valid license key for your copy of VoiceXML browser (for use more than one session), you should run asterisk and get your code by typing the following commands:

```
*CLI> vxml show license
```

The system will give you:

```
Version          : V5.1
Build            :
Gcc              : V4.1
Target           : Pentium i586
Asterisk         : V1.4.22
Date             : Sep 22 2010 18:48:10
Code             : <your_identification_code>
Key              : <your_activation_key>
Max sessions     : 1
```

To order a Commercial License for increase the maximum number of sessions (voicexml ports) supported ,you need to send us <your_identification_code>. Once the code is displayed, copy the code and send it by email to support@i6net.com. Please also include information about your reference installation, such as a purchase order number or the context of the installation.

After payment has been received for the product, I6NET support will send you the license code as soon as possible. Information on how to set up your license will be emailed to you, along with your license key.

To set <your activation key> you must edit the /etc/asterisk/vxml.conf file
I6NET will provide you with an activation that you can update in the license section:

```
[license]
max=100
interface=2
```

```
key=986e211f4e9d3b88fe18f83ace5b65bf
```

max:	maximun number of vxml session that VXI* will manage (it's your ports capactity)
interface:	ethernet interfaces
key:	license key for your server (by default, if the key is empty your system use only one port)

NOTE:

Please keep the emails exchanged with I6NET support, especially the activation key code. It could be necessary to use it again in case of upgrade or reinstallation. With some releases, the upgrade procedure erases all the license information, so the license activation will need to be performed again.
The number of max sessions that can be performed, along with audio and video enablement, will be shown.

5.7 Create your first VoiceXML service

To configure your first VoiceXML service inside Asterisk, just edit your dialpan (extensions.conf):

```
# cd /etc/asterisk
# vi extensions.conf
```

Install first demo VoiceXML examples programs in your /tmp directory:

helloworld1.vxml	; 1st program with TTS “Hello World”” using default EN lenguage
helloworld2.vxml	; 1st program with WAV file access says “Hello World!”
helloworld3.vxml	; 1st program with TTS “Hello World”” using EN-GB language
helloworld.wav	; WAV content file for helloworld2

Add a new extension in your dialplan (/etc/asterisk/extensions.conf) like:

```
exten => 888,1,Answer
exten => 888,2,Wait(3)
exten => 888,3,Vxml(http://vxml.i6net.com/index.vxml)
exten => 888,4,Hangup

exten => 999,1,Answer
exten => 999,2,Wait(3)
exten => 999,3,Vxml(file:///tmp/helloworld2.vxml)
exten => 999,4,Hangup
```

Please ensure you have reloaded your extensions in Asterisk using the prompt command:

```
*CLI> extensions reload
```

Call the service by calling :

```
SIP:888@<your server IP address>
SIP:999@<your server IP address>
```

6 Uninstall or Upgrade

*6.1 Uninstall VXI**

In the same directory as the install.sh script, the uninstall.sh script will remove all installed components and files.

```
# cd vxml_Vx.x_date_build
# ./uninstall.sh
```

*6.2 Update or Upgrade VXI**

If you install a new version over a previously installed version, the installation script retained your configuration file, vxml.conf. Check the new configuration file, vxml.conf.sample, in order to integrate some new parameters into vxml.conf.

Check the VoiceXML interpreter configuration file, client.cfg too.

```
# cd /etc/asterisk
# diff ./vxml.conf ./vxml.conf.sample
# cd /etc/openvxi
# diff ./client.cfg ./client.cfg.sample
```

7 Text-to-Speech (TTS)

The VXI* VoiceXML browser integrates an HTTP client interface to connect to an HTTP text-to-speech (TTS) server. This allows for dynamically generated audio content with a text-to-speech engine. Most VoiceXML browsers have an MRCP (Media Resource Control Protocol) interface to access text-to-speech features. The advantage however, of using the HTTP connector is that the “speech” generated is cached by the VoiceXML Browser, and re-used the next time. The text is posted via an HTTP request then the server responds with a standard wav file. Users can use the TTS to generate menus prompts, for example, without need to purchase a lot of TTS licenses. The integrated TTS packages use PHP, so you need to have an Apache/PHP already installed on your server (MySQL is optional, it allows to produce “CDR” like reporting information to evaluate the TTS using).

List of TTS Supported:

- Acapela TTS
- Baratinoo TTS
- Cepstral TTS
- eSpeak (with/without Mbrola voices) TTS (FREE)
- Flite TTS (FREE)
- Ivona TTS
- Loquendo TTS
- Lumenvox TTS
- Nuance TTS
- Verbio TTS
- VoiceInteraction TTS
- Yantra Software TTS

NOTE:

If you need to install or configure any other TTS engine, please contact our team.

NOTE:

After installing the TTS/HTTP interface you need to configure the VoiceXML browser to use to it :

– add texttospeech=yes in the /etc/asterisk/vxml.cong

– set the corresponding values in the section “TTS server configuration” in /etc/openvxi/client.cfg

7.1 Flite TTS

Vendor

Speech CS CMU

http://www.speech.cs.cmu.edu/flite/

Definition

This is a free open source Text To Speech engine. Flite (Festival Lite) is a small, fast run-time synthesis engine developed at CMU and primarily designed for small embedded machines and/or large servers.

Installation

Unzip and untar the Flite package by using the command:

```
# tar xvzf flite_V1.x_date.tar.gz
```

Go to the directory of the Flite and then, type the following command.

```
# cd flite_V1.x_date
# ./install.sh
```

Configuration

This is the PHP script used to generate wav files from Flite:

```
<?php
error_reporting ( E_ALL );

include("config.php");
include("ttslib.php");

if(!defined('DEBUG'))
  error_reporting(0);

debug("Mode debug enabled");

if(isset($_POST["text"])){
  $text = $_POST["text"];
  if(isset($_POST["format"])) $format = $_POST["format"];
  debug(" POST VARIABLES :");
  debug(print_r( $_POST,true));
}elseif(isset($_GET["text"])){
  $text = $_GET["text"];
  if(isset($_GET["format"])) $format = $_GET["format"];
  debug(" GET VARIABLES :");
  debug(print_r( $_GET,true));
}else
  httperror();

debug(" text   = $text");
debug(" format = $format");

switch($format){
  case 'gif':
    headerdebug("Content-Type: image/gif");
    headerdebug('Content-Disposition: attachment; filename="file.gif"');
    break;
  case 'wav':
  default :
```

```
    headerdebug("Content-Type: audio/wav");
    headerdebug('Content-Disposition: attachment; filename="file.wav"');
}

$filename=tempnam("/tmp",$ttsname."TMP");
$file=fopen($filename, "w");

debug(" Temp file = $filename");

register_shutdown_function('cleanup',"$filename");
fwrite($file, $text);
fwrite($file, "\n");
fclose($file);

$program = "/usr/bin/flite";
$time_start = microtime(true);
register_shutdown_function('cleanup',"$filename.wav");
exec($program." -f $filename -o $filename.wav", $return, $status);
$time_end = microtime(true);
$texec = $time_end - $time_start;
$outsize=filesize("$filename.wav");

readfile("$filename.wav");

if($enable_record_to_cdr)
register_shutdown_function('cdrrecord',$ttsname,$lang,$status,$texec,$text,
$outsize);

register_shutdown_function('garbage',$ttsname);

// vim: set filetype=php expandtab tabstop=2 shiftwidth=2 autoindent
smartindent:
?>
```

The TTS package install the HTTP scripts in : /var/www/tts/flite

In a default Debian installation, you can test the TTS installation with : http://yourip/tts/flite/index.php

This web page will generate wav files in PCM 16bit 8kHz mono.

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Speech Option) and restart the VoiceXML interpreter to get the new configuration.

7.2 Loquendo TTS

Vendor

Loquendo

http://www.loquendo.com

Definition

Loquendo TTS software provides a life-like voice for the dynamic data and prompts in your server-based, multimedia, PDA, embedded and multimodal voice applications.

Installation

First unzip and untar the Loquendo package by using the command:

```
# tar xvzf loquendo_Vx.x_date.tar.gz
```

Go to the directory of the gnudialer; then, type the following command:

```
# cd loquendo_Vx.x_date
# ./install.sh
```

The script asks for an installation directory (default: /opt/LTTS). In this directory you need to enable read/write and execution permissions, otherwise the installation will end prematurely.

Configuration

The TTS package install the HTTP scripts in : /var/www/tts/loquendo

In a default Debian installation, you can test the TTS installation with : http://yourip/tts/loquendo/index.php

This web page will generate wav files in PCM 16bit 8kHz mono.

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Speech Option) and restart the VoiceXML interpreter to get the new configuration.

7.3 Acapela TTS

Vendor

Acapela Group

http://www.acapela-group.com

Definition

Acapela Group gives a voice to all your products and services. We invent voice solutions to give the say to content and enrich user interfaces in a natural and intuitive way based on our comprehensive technologies.

Installation

First unzip and untar the Acapela package by using the command:

```
# tar xvzf acapela_Vx.x_date.tar.gz
```

Next, go to the directory of the Acapela and type the following command:

```
# cd acapela_Vx.x_date
# ./install.sh
```

Configuration

The TTS package install the HTTP scripts in : /var/www/tts/acapela

In a default Debian installation, you can test the TTS installation with : http://yourip/tts/acapela/index.php

This web page will generate wav files in raw alaw or ulaw files.

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Speech Option) and restart the VoiceXML interpreter to get the new configuration.

7.4 Cepstral TTS

Vendor

Cepstral

http://www.cepstral.com

Definition

Cepstral builds high quality, natural sounding voices for server, desktop, and hand-held voice applications. Cepstral also provides professional services to customize and tune voices. Cepstral's high quality text-to-speech voices are available free for trial via the Internet. Cepstral offers 6 different US English voices plus voices for 5 other languages including UK English, Spanish, French, Italian, and German. Cepstral voices are SAPI 5 compliant.

Installation

First unzip and untar the Cepstal package by using the command:

```
# tar xvzf cepstral_Vx.x_date.tar.gz
```

Next, go to the directory of the Cepstral and type the following command:

```
# cd cepstral_Vx.x_date
# ./install.sh
```

Configuration

The TTS package install the HTTP scripts in : /var/www/tts/cepstral

In a default Debian installation, you can test the TTS installation at: http://yourip/tts/cepstral/tts.html

This web page will generate wav files in PCM 16bit 8kHz mono.

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Speech Option) and restart the VoiceXML interpreter to get the new configuration.

7.5 Verbio TTS

Vendor

Verbio

http://www.verbio.com

Definition

Verbio is a company based in Barcelona (Spain), specializing in speech technologies, aimed basically at the Spanish, French, Portuguese and Latin American markets. Contact Verbio directly for more information about how their product works with Asterisk.

You can download the software packages here:

http://www.verbio.com/webverbio2/html/soportecnic.php?tema=download

Installation

If your distribution does not use 'rpm' nor 'deb' please contact Verbio Technologies staff (support@verbio.com) to get specific installation packages or support.

Download the package after contacting the Verbio sales.

```
# dpkg -i verbio-engines-x.yy.deb
# dpkg -i --force-overwrite verbio-clients-x.yy.deb
# dpkg -i --force-overwrite verbio-tts-*-x.yy.deb
# dpkg -i --force-overwrite verbio-asr-*-x.yy.deb (for the ASR)
```

If you have a licence in a USB token, you need to install the USB token driver ('sntl-sud_x.y.z-w_i386.deb'). You will find sntl package in '/usr/share/doc/verbio'.

```
# cd /usr/share/doc/verbio
# dpkg -i sntl-sud_7.3.0-1_i386.deb
```

NOTE:

You need to reboot the server to enable the USB deamon ().

If you have a file licence, please copy it to '/opt/verbio/lic'.

Download and install the I6NET package.

First unzip and untar the verbio package by using the command:

```
# tar xvzf verbio_Vx.x_date.tar.gz
```

Next, go to the directory of the verbio and type the following command:

```
# cd verbio_Vx.x_date
# ./install.sh
```

Configuration

The TTS package install the HTTP scripts in : /var/www/tts/verbio.

In a default Debian installation, you can test the TTS installation at: http://yourip/tts/verbio/tts.html

This web page will generate raw alaw or ulaw files.

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Speech Option) and restart the VoiceXML interpreter to get the new configuration.

8 Text-to-Video (TTV)

The Text-to-Video works much like Text-to-Speech. It can be used to dynamically generate a video, either static or with animation, from text. The text comes from the <prompt> VoiceXML tag and if the attribute language is either “video” or “text.”

Example

```
<prompt bargein="false" xml:lang="text">
Welcome
Camera Service!
</prompt>
```

Screenshot

NOTE:

The HTTP Text-to-Video generates an .h263/.h264 or mp4/3gp file.

Installation

First unzip and untar the text2video package by using the command:

```
# tar xvzf text2video_Vx.x_date.tar.gz
```

Next, go to the directory of the text2video and type the following command:

```
# cd text2video_Vx.x_date
# ./install.sh
```

NOTE:

Install the script on your Apache/PHP server. Configure the VoiceXML Browser (Text-to-Video Option) and restart the VoiceXML interpreter to get the new configuration.

9 Automatic-Speech-Recognition (ASR)

The VXI* VoiceXML Browser uses the Digium/Asterisk speech connector bridge for Lumenvox, Vestec, VoiceInteraction and Verbio Speech engines. All these engines are made for Asterisk platforms. For MRCP compliant ASR engines, you have to install our Xtras* uniMRCP package.

List of Supported ASR engines:

- Loquendo ASR (through MRCP)
- Lumenvox ASR
- Nuance ASR (through MRCP)
- Verbio ASR
- Vestec ASR
- VoiceInteraction ASR

NOTE:

If you need to install or configure any other ASR engine, please contact our team.

9.1 Lumenvox ASR

Vendor

Lumenvox

http://www.lumenvox.com

Definition

The LumenVox Speech Engine is the only speech software that is directly programmed to work with the connector bridge and will allow your Asterisk applications to accept input via a caller's voice. LumenVox supports US, UK and Australian English, Mexican and South American Spanish, and Canadian French.

Installation

Before configuring speech recognition for the VoiceXML browser, install the LumenVox Speech Engine. Complete instructions for downloading and installing LumenVox on Asterisk can be found here: http://www.lumenvox.com/help/speechEngineAsterisk/index.htm.

Following are the main steps for a Debian distribution installation:

Add in /etc/apt/sources.list

```
deb http://www.lumenvox.com/packages/Debian binary/
```

Install the following packages:

```
apt-get install lumenvoxcore
atg-get install lumenvoxclient
atg-get install lumenvoxsre
apt-get install lumenvoxlicenseserver
```

Start LumenVox deamons.

```
/etc/init.d/lvlicensed start
/etc/init.d/lvsred start
```

Generate the Server ID File.

```
/usr/bin/lv_license_manager -g /root/Info.bts
```

Upload it in the LumenVox server with your account, and download the license.

Install the license.

```
/usr/bin/lv_license_manager -m /root/Licensexxxx_xx.bts
```

Download and install the Asterisk 1.4 connector bridge (tar.gz).

```
cp res_speech_lumenvox.so /usr/lib/asterisk/modules/
cp lumenvox.conf /etc/asterisk/
```

Lastly, restart Asterisk.

Configuration

To enable speech recognition, change the main speech parameter.

The “speech” parameter can get three values, “yes”, “automatic”, “no” or “emulation” (don’t generate errors if you enable speech grammars).

```
[general]
…
speech=automatic
speechprovider=lumenvox
…
```

You can specify the path directory where the built in grammars are stored with the “speechdirectory” parameter. The default value is “/opt/lumenvox/engine/vxi/%s/%s.gram.”

With the %s markers you can specify two parameters:

* Language parameters (examples: “es-ES” or “en-UK”)

* Grammar name type, from the field attribute, “type” (examples: “digits”, “number”)

```
[general]
…
speechdirectory=/opt/lumenvox/engine/vxi/%s/%s.gram
…
```

In the VoiceXML browser configuration file :

```
############################
# ASR server configuration #
############################

client.rec.resource.0.cacheDir              VXIString   /tmp/cacheContent
client.rec.resource.0.format                VXIString   grxml
```

You should create the grammar files or create symbolic links to the default grammars installed with the LumenVox package.

```
cd /opt/lumenvox/engine
mkdir vxi
cd vxi
mkdir en-UK
ln -s ../../Lang/BuiltinGrammars/ABNFNumber.gram number.gram
ln -s ../../Lang/BuiltinGrammars/ABNFDigits.gram digits.gram
ln -s ../../Lang/BuiltinGrammars/ABNFPhone.gram phone.gram
```

The following VoiceXML example uses the speech recognition, with the built in grammar, ‘digits.’

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-UK">
 <form>
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <field name="text" type="digits">
   <catch event="noinput nomatch">
    <reprompt/>
   </catch>
   <prompt>
    Speak to me:
   </prompt>
  </field>
  <filled>
   <prompt>
    You say me:
    <value expr="text" />
   </prompt>
   <clear namelist="text" />
  </filled>
```

```
 </form>
</vxml>
```

NOTE:

To enable dynamic grammars you need to configure the VXI* VoiceXML browser to generate temporary grammar files (look at our VXI* Developer Guide).

9.2 Verbio ASR

Vendor

Verbio

http://www.verbio.com

Definition

Verbio is a company based in Barcelona (Spain), specializing in speech technologies, aimed basically at the Spanish, French, Portuguese and Latin American markets. Contact Verbio directly for more information about how their product works with Asterisk.

You can download the software packages here:

http://www.verbio.com/webverbio2/html/soportecnic.php?tema=download

Installation

If your distribution does not use 'rpm' nor 'deb' please contact Verbio Technologies staff (support@verbio.com). Download the package after contacting the Verbio sales.

```
# dpkg -i verbio-engines-x.yy.deb
# dpkg -i --force-overwrite verbio-clients-x.yy.deb
# dpkg -i --force-overwrite verbio-tts-*-x.yy.deb  (for the TTS only)
# dpkg -i --force-overwrite verbio-asr-*-x.yy.deb
```

If you have a licence in a USB token, you need to install the USB token driver ('sntl-sud_x.y.z-w_i386.deb'). You will find sntl package in '/usr/share/doc/verbio'.

```
# cd /usr/share/doc/verbio
# dpkg -i sntl-sud_7.3.0-1_i386.deb
```

NOTE:

You need to reboot the server to enable the USB deamon ().

If you have a file licence, please copy it to '/opt/verbio/lic'.

Download and install the I6NET package to load the Asterisk module res_speech_verbio.so.

(This package contents the TTS scripts too.)

First unzip and untar the verbio package by using the command:

```
# tar xvzf verbio_Vx.x_date.tar.gz
```

Next, go to the directory of the verbio and type the following command:

```
# cd verbio_Vx.x_date
# ./install.sh
```

Configuration

To enable speech recognition, change the main speech parameter.

The “speech” parameter can get three values, “yes”, “automatic”, “no” or “emulation” (don’t generate errors if you enable speech grammars).

```
[general]
…
speech=automatic
speechprovider=verbio
…
```

Don't set the speechdirectory parameter.

In the VoiceXML browser configuration file :

```
############################
# ASR server configuration #
############################

client.rec.resource.0.cacheDir              VXIString   /tmp/cacheContent
client.rec.resource.0.format                VXIString   txt
```

The following VoiceXML example uses the speech recognition, with the built in grammar, ‘digits.’

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-UK">
 <form>
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <field name="text" type="digits">
   <catch event="noinput nomatch">
    <reprompt/>
   </catch>
   <prompt>
    Speak to me:
   </prompt>
  </field>
  <filled>
   <prompt>
    You say me:
    <value expr="text" />
   </prompt>
   <clear namelist="text" />
  </filled>
 </form>
</vxml>
```

9.3 VoiceInteraction ASR

Vendor

VoiceInteraction

http://www.voiceinteraction.pt

Definition

VoiceInteraction, founded in 2008 and based in Lisboa (Portugal), is a company specialised in the development of speech technologies. VoiceInteraction develops voice synthesis and speech recognition engines for many web and telephony applications.

Installation

This is for de Debian Lenny version, for other Linux, read the official Audimus Installation manual.

To configure the apt client, just edit and add to the file /etc/apt/sources.list one of the following line set.

For asterisk 1.4

```
deb http://services.voiceinteraction.pt/repo/Debian/5.0 engines 3rdparty Dixi
Audimus
deb http://services.voiceinteraction.pt/repo/Debian/5.0 asterisk.1.4 Dixi
Audimus
```

For asterisk 1.6.1.x

```
deb http://services.voiceinteraction.pt/repo/Debian/5.0 engines 3rdparty Dixi
Audimus
deb http://services.voiceinteraction.pt/repo/Debian/5.0 asterisk.1.6.1 Dixi
Audimus
```

For asterisk 1.6.2

```
deb http://services.voiceinteraction.pt/repo/Debian/5.0 engines 3rdparty Dixi
Audimus
deb http://services.voiceinteraction.pt/repo/Debian/5.0 asterisk.1.6.2 Dixi
Audimus
```

Refresh the package lacal base with :

```
# apt-get update
```

And Install the package audimus-asterisk-xx-xx (where xx-xx is the language requested)

```
# apt-get install audimus-asterisk-es-es
```

The following extra packages will be installed:

audimus audimus-config audimus-model-es-es-monophones-g2p-phonemodels audimus-model-es-es-monophones-mlp-telephone audimus-model-es-es-monophones-task-asterisk

Activate the license with:

```
# audimus_activate_license
```

Please enter your Audimus license:

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Your License was successfully activated!
```

Configuration

To enable speech recognition, change the main speech parameter.

The “speech” parameter can get three values, “yes”, “automatic”, “no” or “emulation” (don’t generate errors if you enable speech grammars).

```
…
speech=automatic
speechprovider=verbio
…
```

In the VoiceXML browser configuration file:

```
############################
# ASR server configuration #
############################

client.rec.resource.0.cacheDir              VXIString   /tmp/cacheContent
client.rec.resource.0.format                VXIString   txt
client.rec.resource.0.syntax                VXIString   doctype
```

You need to restart the VXI* and Asterisk to get all the changes.

Logs file from the ASR engine:

```
# tail -f /var/log/VI/VI.log
```

The following VoiceXML example uses the speech recognition, with the built in grammar, 'digits.'

```
<?xml version="1.0" encoding="iso-8859-1"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml" xml:lang="en-UK">
 <form>
  <property name="inputmodes" value="voice"/>
  <property name="timeout" value="30s"/>
  <field name="text" type="digits">
   <catch event="noinput nomatch">
    <reprompt/>
   </catch>
   <prompt>
    Speak to me:
   </prompt>
  </field>
  <filled>
   <prompt>
    You say me:
    <value expr="text" />
   </prompt>
   <clear namelist="text" />
  </filled>
 </form>
</vxml>
```

9.4 Vestec ASR (over MRCP)

Vendor

Vestec

http://www.vestec.com

Definition

Vestec Speech Engine has been designed to offer a robust yet highly affordable solution for enabling speech recognition with Asterisk. Vestec MRCP engine is available online on thier web site, you can purchase an speech starter kit (for 45$).

Generate HostID

Before buying a license, you need to generate a Host ID number from your server.

Download the tool to generate the Host Id, execute it on your server to get the Host ID.

```
root:~/vestec# ./generate_host_id.Debian.602_i386
Your Host id is e0979faac77437b54bfa6fa50bd34fea
```

Installation

Buy the starter kit (ASR engine + ASR language packages).

You will receive a mail with a link to download the Vestec packages.

For Debian, (use apt-get -install to resolve the dependencies) :

```
vasre-lang-*
vasre-so*
vasre-tools*
vasre-rm*
vasre-server*
vasre-client*
vasre-mrcp*
```

Register your installation, obtain the license file, and copy it to /opt/Vestec/license/..

```
# cp license.lic /opt/Vestec/license/
```

Edit the Vestec MRCP configuration file (/opt/Vestec/mrcp/conf/mrcpserver.xml), and set the IP/interface for the MRCP connector :

```
  <properties>
    <!-- By default, network interface (IP address) to bind to will be
implicitly identified.
         Alternatively, it can be explicitly set.
    -->
    <ip type="auto"/>
    <ip>192.168.200.109</ip>
    <!-- <ext-ip>a.b.c.d</ext-ip> -->
  </properties>
```

Start the services:

```
# /etc/init.d/vasre-rm start
# /etc/init.d/vasre-server start
# /etc/init.d/vasre-mrcp start
```

Configuration

To enable speech recognition, change the main speech parameter.

The “speech” parameter can get three values, “yes”, “automatic”, “no” or “emulation” (don’t generate errors if you enable speech grammars).

```
…
speech=automatic
speechprovider=unmrcp
…
```

In the VoiceXML browser configuration file:

```
############################
# ASR server configuration #
############################

client.rec.resource.0.cacheDir              VXIString   /tmp/cacheContent
```

```
client.rec.resource.0.format                VXIString   grm
```

You need to restart the VXI and Asterisk to get all the changes.

Logs files from the ASR engine are generated here:

```
# ls /var/log/VestecASRE
```

9.5 Configure Safe Asterisk/openvxi

Safe_asterisk is a script that runs asterisk in a loop, which can be useful if you fear asterisk may crash.

The script does not run in the background like a standard service. Rather, it runs in its own linux virtual console (9, by default). It also uses the option '-c' of asterisk(8) to avoid detaching asterisk from that terminal.

safe_asterisk also runs asterisk with unlimited core file size, and thus asterisk will dump core in case of a crash.

To get a "picture" of console 9, from another terminal (e.g: from a remote shell session) you can use:

screendump 9

The init script of the Debian package should be able to run safe_asterisk as the asterisk service, if so configured. See coments in /etc/default/asterisk

If Asterisk crashes, the VoiceXML browser needs to be restarted before restarting the Asterisk (in order to clear the allocated sessions and resources).

Edit and add this line in the script /usr/sbin/safe_asterisk

```
#NOTIFY=ben@alkaloid.net    # Who to notify about crashes
#EXEC=/path/to/somescript   # Run this command if Asterisk crashes
EXEC="/etc/init.d/openvxi restart"
MACHINE=`hostname`          # To specify which machine has crashed when getting
the mail
```

10 Configuration Files (examples)

Find here the most important configuration files examples provided for our packages.

10.1 client.cfg

```
# ===========================================================================
#
# Copyright (c) 2001-3 SpeechWorks International
#
# Configuration file for SpeechBrowser client application
#
# Rules:
# - Lines beginning with '#' are considered comments and ignored
# - No comments are supported within a line (following other items)
# - Each line is made of three items, a Name, a Type and a Value
# - These three items (strings) can be separated by tabs or spaces
# - Only the Value string can contain spaces, except trailing spaces
#         which are ignored
# - All items are case-sensitive
# - Supported types are 'Environment', 'VXIString', 'VXIInteger',
#         'VXIFloat' and 'VXIPtr'
# - The 'Environment' type is used to set and remove an environment
#         variable
# - Types other than 'Environment' indicate you want to set a map
#         property
# - All properties will be passed as a single map argument to SB
#         functions
# - The value for the 'VXIPtr' type is ignored and considered as NULL
# - Environment variables set here will apply not only to the script
#         environment, but to the real application as well
# - To remove a variable from the environment, supply no Value for it
# - To use a variable within the script, use the syntax '$(VARIABLENAME)'
# - Variables can only be used within Value items, not in Names and Types
#
# Examples:
#                                 Name                 Type       Value
#                           -----------------------------------------------
#  Set an integer property:  myModule.myIntegerKey   VXIInteger  1234
#  Set a string property:    myModule.myStringKey    VXIString   Any string
#  Set an env. variable:     MY_VARIABLE             Environment C:\TEMP;D:\
#  Remove an env. variable:  MY_EX_VARIABLE          Environment
#  Use an env. variable:     myModule.myEnvKey       VXIString   $(MY_VARIABLE)
#
#  You can use several variables within a Value:  $(TYPE)://$(DRIVE)/$(PATH)
#
# ===========================================================================

####################################
# Overridden environment variables #
####################################

# SWISBSDK        Environment C:\Progra~1\SpeechWorks


#############################
# Base client configuration #
#############################
```

```
########################################
### REC, PROMPT and TEL implementation #
########################################

client.rec.implementation                    VXIString   voicebrowser
client.tel.implementation                    VXIString   voicebrowser
client.prompt.implementation                 VXIString   voicebrowser

### Inet and Write-back cache
client.cache.cacheDir                        VXIString   /tmp/cacheContent
client.cache.cacheTotalSizeMB                VXIInteger  200
client.cache.cacheLowWaterMarkMB             VXIInteger  180
client.cache.cacheEntryMaxSizeMB             VXIInteger  20
client.cache.cacheEntryExpTimeSec            VXIInteger  3600
client.cache.unlockEntries                   VXIInteger  1

### Logging
client.log.filename                          VXIString   /tmp/log.txt
client.log.fileMimeType                      VXIString   text/plain;charset=ISO-
8859-1
#client.log.fileMimeType                      VXIString   text/plain;charset=UTF-
8
client.log.maxLogSizeMB                      VXIInteger  50
client.log.contentDir                        VXIString   /tmp/logContent
client.log.contentTotalSizeMB                VXIInteger  50
# The default is to log to standard out as well as to a file (set to 1)
# set to 0 to disable logging to standard out
client.log.logToStdout                       VXIInteger  0
# The default is to keep the log file open between writes for faster
# logging (set to 1), set to 0 to close between writes to allow
# manually rotating the log file by merely moving it aside while the
# platform continues running
client.log.keepLogFileOpen                   VXIInteger  1
# The default is to report the error text for each error, as contained
# in the XML error mapping files defined below
client.log.reportErrorText                   VXIInteger  1

### Internet fetch, extension rules defined separately below
#client.inet.proxyServer                      VXIString   172.17.100.1
#client.inet.proxyPort                        VXIInteger  8080
client.inet.userAgent                        VXIString   OpenVXI/3.1
client.inet.acceptCookies                    VXIInteger  1
client.inet.postContinueTimeout              VXIInteger  1

### Proxy rules
#client.inet.proxyRule.0                      VXIString
.speechworks.com/specialPath | proxyServer1:123
#client.inet.proxyRule.1                      VXIString   grenoble.ferma |
#client.inet.proxyRule.2                      VXIString   .com | proxyServer2:456
#client.inet.proxyRule.3                      VXIString   | proxyServer3:789

### JavaScript
client.jsi.runtimeSizeBytes                  VXIInteger  16384000
client.jsi.contextSizeBytes                  VXIInteger  131072
client.jsi.maxBranches                       VXIInteger  100000
#client.jsi.globalScriptFile                  VXIString
http://greenland/misc/test.js

### QA Prompt cache
client.prompt.enableCache                    VXIInteger  1

### Session connection variables
client.session.connection.local.uri          VXIString   telephone.dnis
client.session.connection.remote.uri         VXIString   telephone.ani
```

```
client.session.connection.protocol.name     VXIString   OpenVXI_VXML
client.session.connection.protocol.version  VXIString   1.0.0
client.session.connection.aai               VXIString   VXML Application
client.session.connection.originator        VXIString   connection.remote
# redirect array: each element of the array has 4 properties:
# uri, pi, si and reason
client.session.connection.redirect.0.uri    VXIString
http://www.company1.com/redirect
client.session.connection.redirect.0.pi     VXIString   presentation information
0
client.session.connection.redirect.0.si     VXIString   screening information 0
client.session.connection.redirect.0.reason VXIString   unknown
client.session.connection.redirect.1.uri    VXIString
http://www.company2.com/redirect
client.session.connection.redirect.1.pi     VXIString   presentation information
1
client.session.connection.redirect.1.si     VXIString   screening information 1
client.session.connection.redirect.1.reason VXIString   unknown

### TRD utilities
# The stack size in bytes to use when creating new threads. If set to
# zero or left undefined it means 'use the default (OS-specific) size',
# which will usually be the same stack size as the parent process.
#client.trd.threadStackSize                  VXIInteger  0

### SSFT's Recognizer configuration
#client.rec.initURI                          VXIString   $
(SWISRSDK)/config/Baseline.xml

### VoiceXML Interpreter
client.vxi.beepURI                          VXIString   uri:beepbis.gsm
# Uncomment the following to override the interpreter defaults
#client.vxi.defaultsURI                      VXIString
file:///etc/openvxi/defaults.xml
#client.vxi.maxLoopIterations                VXIInteger 50
#client.vxi.maxDocuments                     VXIInteger 20

#################################################
# Base diagnostic tag offset for each interface #
#################################################

client.cache.diagLogBase                    VXIInteger  2000
client.inet.diagLogBase                     VXIInteger  3000
client.jsi.diagLogBase                      VXIInteger  4000
client.prompt.diagLogBase                   VXIInteger  5000
client.rec.diagLogBase                      VXIInteger  6000
client.tel.diagLogBase                      VXIInteger  7000
client.vxi.diagLogBase                      VXIInteger  8000
client.object.diagLogBase                   VXIInteger  9000
client.client.diagLogBase                   VXIInteger  10000


##############################################
# Diagnostic tags: 0 to disable, 1 to enable #
##############################################

# API/general log traces for each component
client.log.diagTag.0                        VXIInteger  0
client.log.diagTag.2000                     VXIInteger  0
client.log.diagTag.2001                     VXIInteger  0
client.log.diagTag.2002                     VXIInteger  0
client.log.diagTag.3000                     VXIInteger  0
client.log.diagTag.3001                     VXIInteger  0
client.log.diagTag.3002                     VXIInteger  0
```

```
client.log.diagTag.3003                     VXIInteger  0
client.log.diagTag.3004                     VXIInteger  0
client.log.diagTag.3005                     VXIInteger  0
client.log.diagTag.3006                     VXIInteger  0
client.log.diagTag.3007                     VXIInteger  0
client.log.diagTag.3008                     VXIInteger  0
client.log.diagTag.3010                     VXIInteger  0
client.log.diagTag.4000                     VXIInteger  0
client.log.diagTag.4001                     VXIInteger  0
client.log.diagTag.4002                     VXIInteger  0
client.log.diagTag.5000                     VXIInteger  0
client.log.diagTag.5001                     VXIInteger  0
client.log.diagTag.6000                     VXIInteger  0
client.log.diagTag.6001                     VXIInteger  0
client.log.diagTag.6002                     VXIInteger  0
client.log.diagTag.7000                     VXIInteger  0
client.log.diagTag.9000                     VXIInteger  0

# VXI logging, the first is for application diagnostics/errors, the
# second is the output from the <log> element
client.log.diagTag.8000                     VXIInteger  0
client.log.diagTag.8001                     VXIInteger  0
client.log.diagTag.8002                     VXIInteger  0
client.log.diagTag.8003                     VXIInteger  0

# SBclient API, component, and generic logging respectively
client.log.diagTag.10000                    VXIInteger  0
client.log.diagTag.10001                    VXIInteger  0
client.log.diagTag.10002                    VXIInteger  0

# testClient logging
client.log.diagTag.60001                    VXIInteger  0

# Pass-through of OpenSpeech Recognizer diagnostic messages as enabled
# in the configured OSR diagnostic tag map file, always leave this
# enabled
client.log.diagTag.79999                    VXIInteger  0

#######################
# Error mapping files #
#######################

client.log.errorMapFile.1                   VXIString
/usr/lib/openvxi/VXIclientErrors.xml
client.log.errorMapFile.2                   VXIString
/usr/lib/openvxi/VXIErrors.xml
client.log.errorMapFile.3                   VXIString
/usr/lib/openvxi/VXIobjectErrors.xml
client.log.errorMapFile.4                   VXIString
/usr/lib/openvxi/VBpromptErrors.xml
client.log.errorMapFile.5                   VXIString
/usr/lib/openvxi/VBrecErrors.xml
client.log.errorMapFile.6                   VXIString
/usr/lib/openvxi/VBtelErrors.xml
client.log.errorMapFile.7                   VXIString
/usr/lib/openvxi/SBcacheErrors.xml
client.log.errorMapFile.8                   VXIString
/usr/lib/openvxi/SBinetErrors.xml
client.log.errorMapFile.9                   VXIString
/usr/lib/openvxi/SBjsiErrors.xml
client.log.errorMapFile.10                  VXIString
/usr/lib/openvxi/SBlogErrors.xml

############################
```

```
# TTS server configuration #
############################

#client.prompt.resource.0.uri                VXIString
http://localhost/tts/flite/tts.php
#client.prompt.resource.0.uriVideo           VXIString
http://localhost/tts/video/ttv.php
client.prompt.resource.0.method             VXIString   POST
client.prompt.resource.0.cacheDir           VXIString   /tmp/cacheContent
client.prompt.resource.0.format             VXIString   wav
client.prompt.resource.0.formatVideo        VXIString   h263
client.prompt.resource.0.maxage             VXIInteger  -1

############################
# ASR server configuration #
############################

client.rec.resource.0.cacheDir              VXIString   /tmp/cacheContent
client.rec.resource.0.format                VXIString   bnf

#############################################
# File extension to MIME type mapping rules #
#############################################

client.inet.extensionRule.xml               VXIString   text/xml
client.inet.extensionRule.txt               VXIString   text/plain
client.inet.extensionRule.ulaw              VXIString   audio/basic
client.inet.extensionRule.wav               VXIString   audio/x-wav
client.inet.extensionRule.alaw              VXIString   audio/x-alaw-basic
client.inet.extensionRule.srgs              VXIString   application/srgs+xml
client.inet.extensionRule.grxml             VXIString   application/srgs+xml
client.inet.extensionRule.grm               VXIString   application/isolated
client.inet.extensionRule.jsgf              VXIString   application/x-jsgf
client.inet.extensionRule.bnf               VXIString   application/verbio-abnf
client.inet.extensionRule.gram              VXIString   application/lumenvox-
abnf
client.inet.extensionRule.ssml              VXIString
application/synthesis+ssml
client.inet.extensionRule.vxml              VXIString   application/voicexml+xml

###########################
# Objects initialisations #
###########################

#client.object.initialisation.demo           VXIString   UP
```

10.2 vxml.conf

```
;
; VoiceXML Configuration
;

[general]
autoanswer=yes
video=no
speech=emulation
wavcodec=gsm
videosilence=
audiosilence=
dialformat=SIP/%s
speechprovider=lumenvox
speechscore=500

[control]
forward=#
reverse=*
stop=123456789
pause=
restart=0
skipms=5000

[license]
max=1
key=

[account0]
name=demos
number=*0
url=http://links.i6net.com/vxml/index.vxml
max=1
speech=no
dialformat=SIP/%s@myaccount

[account1]
name=audiovideo
number=*1
url=http://localhost/vxml/audio.vxml
urlvideo=http://localhost/vxml/video.vxml
max=1
speech=no
dialformat=SIP/%s
```

10.3 extensions.conf

```
;
; Static extension configuration file, used by
; the pbx_config module. This is where you configure all your
; inbound and outbound calls in Asterisk.
;

;
; The "General" category is for certain variables.
;
[general]
static=yes
writeprotect=no


[globals]
CONSOLE=Console/dsp                         ; Console interface for demo
TRUNK=sip                            ; Trunk interface
TRUNKMSD=1                           ; MSD digits to strip (usually 1 or 0)
URL=file:///root/index.vxml


[default]

; Echo test
exten => 600,1,Wait,1
exten => 600,2,Answer
exten => 600,3,BackGround(beepbis)
exten => 600,4,Echo

; Netann support
exten => dialog,1,Answer
exten => dialog,2,Vxml(${URL})
exten => dialog,3,Hangup

; SIP users
exten => _userX,1,Answer()
exten => _userX,2,Dial(SIP/${EXTEN})

; H324 hosting
exten => _.,1,Set(__VXML_NUMBER=${EXTEN})
exten => _.,2,Goto(default,vxml,1)

exten => vxml,1,NoOp(${CHANNEL(transfercapability)})
exten => vxml,n,Set(__VXML_ID=${UNIQUEID})
exten => vxml,n,GotoIf($[${CHANNEL(transfercapability)}=VIDEO]?video:digital)
exten => vxml,n(digital),GotoIf($[${CHANNEL(transfercapability)}=DIGITAL]?
video:audio)
exten => vxml,n(video),Answer()
exten => vxml,n,NoCDR()
exten => vxml,n,h324m_gw(vxml_gw@default/n)
exten => vxml,n,Hangup()
exten => vxml,n(audio),Answer()
;exten => vxml,n,Set(CALLERID(number)=${CALLERID(number):-4})
exten => vxml,n,Wait(1)
exten => vxml,n,Vxml(${VXML_NUMBER})
exten => vxml,n,Hangup()
exten => vxml_gw,1,Answer()
;exten => vxml_gw,n,Set(CALLERID(number)=${CALLERID(number):-4})
exten => vxml_gw,n,h324m_gw_answer()
exten => vxml_gw,n,Set(CHANNEL(transfercapability)=VIDEO)
```

```
exten => vxml_gw,n,Wait(4)
exten => vxml_gw,n,Vxml(${VXML_NUMBER})
exten => vxml_gw,n,Hangup()

exten => dial3g,1,h324m_call(dial3g_gw@default)
exten => dial3g_gw,1,Set(CHANNEL(transfercapability)=VIDEO)
exten => dial3g_gw,n,Dial(ZAP/g1/${VXML_NUMBER})
```

10.4 sip.conf

```
;
; SIP Configuration
;

[general]
disallow=all
allow=h263p
allow=alaw
allow=ulaw
port=5060
videosupport=yes
defaultexpirey=120
dtmfmode=auto

;register => username:password@ip/phone

[peer-out]
type=peer
secret=password
username=username
host=localhost
dtmfmode=inband
canreinvite=no
context=default

[peer-in]
type=peer
dtmfmode=inband
canreinvite=no
context=default
host=localhost

[user1]
type=peer
username=user1
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
nat=yes

[user2]
type=peer
username=user2
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
nat=yes

[user3]
type=peer
username=user3
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
```

```
nat=yes

[user4]
type=peer
username=user4
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
nat=yes

[user5]
type=peer
username=user5
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
nat=yes

[user6]
type=peer
username=user6
allow=h263p
secret=1234
host=dynamic
canreinvite=yes
context=default
nat=yes
```