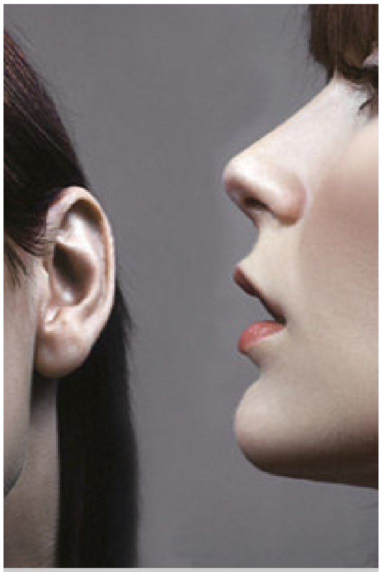

"Offering **integrated** Voice, Web & Mobile Solutions & Services which make communication more effective & more efficient"

VoiceXML enhancements

Comsys SpeechFrame®

Services & Solutions for Voice Web Mobile

COMSYS

Reference number:
Soesterberg, May. 23 2008

This document is the copyright of

Comsys Telecom & Media B.V.

All rights reserved.

© Copyright
Proprietary information of Comsys Telecom & Media B.V. or its affiliates is contained herein. Any reproduction, use, appropriation, or disclosure of this information, in whole or in part, without the specific prior written authorisation of the owners thereof is strictly prohibited. Failure to observe this notice may result in legal proceedings or liability for resulting damage or loss.

Management Summary

Additions to the Comsys SpeechFrame Voice-XML service to support valuable features on the telephony layer. This is all within the Voice-XML standards.

Document management

Version	Date	Description	Author
1.0.0	2008-05-23	First concept	Comsys Telecom & Media B.V. Ronald Elderhorst

Table of contents

1. The <audio> tag

The way how audio fragments are played back can be altered through adding extra 'query' parameters to the src.
This will play back 'welcome.wav' in a normal way

```
...
    <prompt>
        <audio src="welcome.wav/>
    </prompt>
...
```

This will play back 'welcome.wav' in an adapted way

```
...
    <prompt>
        <src="welcome.wav?volume=10;speed=0.5/>
    </prompt>
...
```

The query parameters that can be added are:

- The offset.
  Offset in msec. where play back begins;
- The volume.
  The volume denotes the amplification and is in the range -50 -- +50.
  The actual volume is depends upon the hardware in use;
- The automaticGainEnabled.
  Whether or not to adapt the volume automatically to environmental noise;
- The speed.
  The play back speed from 0.1 – 10.0.
  The actual speed depends upon the hardware in use.

If play done or stopped, the amount of msec. that is played back is returned by the hardware layer.

2. The <record> tag

The way how utterances are recorded can be altered through set additional properties.
This will start a recording in a normal way

```
...
    <record name=”recording” beep=”true” maxtime=”10s” finalsilence=”4s” dtmfterm=”false”/>
...
```

This will start a recording in an adapted way

```
...
    <record name=”recording” beep=”true” maxtime=”10s” finalsilence=”4s” dtmfterm=”false”>
        <property name=”record.silenceElimination” value=”1000”/>
        <property name=”record.threshold” value=”0”/>
        <property name=”record.automaticGainEnabled” value=”true”/>
        <property name=”record.volume” value=”0”/>
    </record>
...
```

The query parameters that can be added are:

- The silenceElimination.
  The initial silence is cut off to silenceElimination msec. at the most.
  This guarantees that the start of the utterance is included in the recording;
- The threshold.
  The level above which the recording is started;
- The automaticGainEnabled.
  Whether or not to adapt the volume automatically to environmental noise;
- The volume.
  The volume denotes the amplification of the recorded utterance and is in the range -50 -- +50.
  The actual volume is depends upon the hardware in use.

If recording is done or stopped, the amount of msec. that is recorded is returned by the hardware layer.
Note that the extra properties are not taken into account yet.

3. The <transfer> tag with enhanced call setup details

The transfer can be used for a variety of functions:

- Normal transfer.
  Only the dnis is taken into account;
- Enhanced transfer.
  This enables control of the ani and rdnis;
- Conference transfer.
  This enables a variety of conference possibilities.

All transfer actions take an <address> as the 'dest' (or 'destexpr') attribute.

An address = <number>[?<option>[;<option>]*]

A number = a telephone number

An option = one of:

- NumberPlan
  { UNKNOWN, ISDN, DATA, TELEX, NATIONAL_STANDARD, PRIVATE }
- NumberFormat
  { UNKNOWN, INTERNATIONAL, NETWORK_SPECIFIC, SUBSCRIBER_NUMBER, ABBREVIATED_NUMBER }
- NumberScreening
  { UNKNOWN, USER_NOT_SCREENED, USER_VERIFIED_PASSED, USER_VERIFIED_FAILED, NETWORK_PROVIDED }
- NumberPresentation
  { UNKNOWN, ALLOWED, RESTRICTED }

If the transfer is done or stopped, the amount of msec. the call was transferred (without a call setup time) is returned.

3.1. Normal transfer to another destination

This will transfer the call in a normal way

```
...
    <transfer name="transfer" bridge="true" dest="<address>"/>
...
```

3.2. Enhanced transfer to another destination

This will start a transfer with enhanced capabilities.

```
...
    <transfer name="transfer" bridge="true"
dest="ani:<address>&amp;dnis:<address>&amp;rdnis:<address>"/>
...
```

The dnis is mandatory while the ani and rdnis are optional.
If the ani is not given the ani as received with the incoming call is used.
An empty ani (ani:) means that no ani is sent in the transfer even when it is known to the system.

4. The <transfer> tag and prompt on b-leg

If, in a transfer, it is required to play back a message to the called party (the b-leg) as soon as this party answers the call, the transfer looks like this:

```
...
    <transfer name="transfer" bridge="true" dest="<address>">
        <prompt>
            <audio src="../prompts/tune.wav?legId=1"/>
        </prompt>
    </transfer>
...
```

Note:

- Only one prompt is queued for play back on the b-leg;
- As the VoiceXmlService requires all prompts to finish before the transfer is started, the 'play done' for the b-leg is given immediately after invocation of the play.
  The actual playing is postponed until the b-leg is answered;
- Due to latency in the signalling path, it can happen that the calling party (a-leg) hears (a small) part of the dialogue that is started by the called party before the called party actually hears the prompt.
  If the called party knows a prompt is played back, it can decide to say nothing until the prompt is played back.

5. Conferencing

Conferencing is implemented through use of the Voice-XML transfer tag.

```
...
    <transfer name="conference" bridge="false" dest="<conference address>"/>
...
```

The most convenient way to administer conferences is the use of a database to construct the <conference address>.

A conference address	= <number>?box=<box name>;mode=<Mode>;exit=<exit value>[;<option>]*
A number	= a telephone number
A box name	= the conference box name
The Mode	= one of {full, listen, talk }
The exit value	= a DTMF that has to be pressed to leave the conference.
An option	= one of:

- application
  To start a dummy inbound Vxml session.

Conference boxes are created and destroyed automatically.
Conferences are split in simple and complex ones.

5.1. A simple conference

Simple conferences have two or more applicants in a box without the possibility to record the conference or play back some prompt in the conference (an applicant is placed in a conference through use of the transfer tag so its Voice-XML session is paused until that stopped).
The following example will:
- Place the applicant in box 'CB123'
- Allow each applicant to speak and to listen
- Leave the conference by pressing a '7'

```
...
    <transfer name="theConference" bridge="false" destexpr="'000?box=CB123;mode=full;exit=7">
        ...
        ...
    </transfer>
...
```

5.2. A complex conference

Complex conferences have one or two dummy Voice-XML sessions started which can be instructed to playback a prompt, record the conference etc.
The following example will:
- Place the applicant in box 'CB123'
- Allow each applicant to speak and to listen
- Leave the conference by pressing a '7'
- Start a dummy Voice-XML session for the (fake) DDI 4567

This enables recording the dialogue OR play back prompts in the dialogue.
For best flexibility, the dialogue is instructed through use of a database.

```
...
    <transfer name="theConference" bridge="false"
        destexpr="'0000?box=CB123;mode=full;exit=7;application=4567">
        ...
```

```
        ...
    </transfer>
...
```

The following example will:

- Place the applicant in box 'CB123'
- Allow each applicant to speak and to listen
- Leave the conference by pressing a '7'
- Start a dummy Voice-XML session for the (fake) DDI 4567
- Start an inbound Voice-XML session for the dummy outbound 1289

This enables recording the dialogue AND play back prompts in the dialogue at one and the same time.
For best flexibility, these dialogues are instructed through use of a database.

```
...
    <transfer name="theConference" bridge="false"
        destexpr="'1289?box=CB123;mode=full;exit=7;application=4567">
        ...
        ...
    </transfer>
...
```

Note that the extra Voice-XML session(s) is (are) terminated when either a session itself terminates or the last applicant leaves the conference.

5.3. A complex conference with outbound

NOT YET IMPLEMENTED.

For this conference an operator is called automatically as one enters the conference box.
The following example will:

- Place the applicant in box 'CB123'
- Allow each applicant to speak and to listen
- Leave the conference by pressing a '7'
- Start a dummy Voice-XML session for the (fake) DDI 4567
- Start an inbound Voice-XML session for the dummy outbound 1289
- Start an outbound call to 1289 and, as soon as the call is answered, connects the call to the previous inbound session.

This enables recording the dialogue AND play back prompts in the dialogue at one and the same time.
For best flexibility, these dialogues are instructed through use of a database.
DTMFs pressed by the operator are sent through the Voice-XML session.

```
...
    <transfer name="theConference" bridge="true"
        destexpr="'1289?box=CB123;mode=full;exit=7;application=4567">
        ...
        ...
    </transfer>
...
```

Note that:

- the extra Voice-XML session(s) is (are) terminated when either a session itself terminates or the last applicant leaves the conference; a possible attached operator will be disconnected
- The given exit=7 has no meaning for the operator.

6. Pre-call announcements

This lets you run (part of) the dialogue before the call is connected i.e. no call charge yet.
It enables database lookups, application initialisation, portal issues etc. to take place.
Until the call is connected, the caller will, dependent upon the switch settings, a ringtone as generated by the switch or a ring-back-tone generated by the application.
To enable this:

- Configure the VoiceXmlService component
  Set the 'ConnectOn' from 'IncomingCall' to 'Application'
- Split up the Voice XML dialogue.
  A pre-call phase and a connected-call phase.

6.1. The pre-call phase

This part of the dialogue must have the property 'delayedInboundConnect' set to:

- The value "true"
  This will delay the connect until instructed otherwise;
- The value "AnswerBLeg"
  This will delay the inbound connect until the B-leg has answered the phone i.e. during a transfer.

Due to the way how VXML-interpreters work, care should be taken that the pre-call part is executed (the dummy field ensures this) before the 'connect' is sent as can be seen in the following example:

```
<?xml version='1.0'?>
<!DOCTYPE vxml SYSTEM "/voicexml2-0.dtd">
<vxml version="2.0" application="../root_doc.vxml">
    <property name="delayedInboundConnect" value="true"/>

    <form id="preCall">
        <field name="dummy" type="digits">
            <prompt timeout="1s">
                <audio src="../prompts/tune.wav"/>
            </prompt>
            <noinput>
                <goto next="mainDialogue.vxml"/>
            </noinput>
            <filled>
                <goto next="mainDialogue.vxml"/>
            </filled>
        </field>
    </form>
</vxml>
```

6.2. The connected call phase

This part of the dialogue (the mainDialogue) must not have the property delayedInboundConnect' or has it set to "false" to let the dialogue continue in a connected phase.

Note that:

- Dependent upon the switch, operation can be limited during the pre-call phase:
- DTMFs can be blocked
- The bearer channel (speech path) can be blocked from the caller to the IVR

7. Sending a specific clearing cause

When running a pre-call dialogue it can be useful to disconnect the (incoming) call with a specific clearing cause out of:

- NORMAL
- NUMBER_BUSY
- NO_ANSWER
- NUMBER_UNOBTAINABLE
- NUMBER_CHANGED
- OUT_OF_ORDER
- INCOMING_CALLS_BARRED
- CALL_REJECTED
- CALL_FAILED
- CHANNEL_BUSY
- NO_CHANNELS
- CONGESTION

Carefully take the appropriate clearing cause.

The following illustrates proper use:

```
<?xml version='1.0'?>
<!DOCTYPE vxml SYSTEM "/voicexml2-0.dtd">
<vxml version="2.0" application="../root_doc.vxml">
    <property name="delayedInboundConnect" value="true"/>
    <property name="clearingCause" value="NUMBER_UNOBTAINABLE"/>

    <form id="preCall">
        <field name="dummy" type="digits">
            <prompt timeout="1s">
                <audio src="../prompts/tune.wav"/>
            </prompt>
            <noinput>
                <disconnect/>
            </noinput>
            <filled>
                <disconnect/>
            </filled>
        </field>
    </form>
</vxml>
```

Note that the voice XML dialogue must have a play (or record) to get the property map that includes the 'clearingCause' property as the disconnect tag takes no properties.