



Réf. : MS\_QSG\_2.40\_001A

**MEDIASPEECH®**



# **WEB QUICK START GUIDE AND MODEL UPDATE**

**VERSION 2.40**

**www.vecsys.fr**

**Siège social** : Parc d'activité du pas du lac, 10 bis av Ampère, 78180 Montigny le Bretonneux, France

**Adresse postale** : Parc d'activité du pas du lac, 10 bis av Ampère, 78180 Montigny le Bretonneux, France

**Tél.** : + 33 (0)1 69 29 87 87 • **Fax** : + 33 (0)1 69 07 58 58

**Site Strasbourg** : 37, boulevard Clemenceau • Bâtiment 6 • 67000 Strasbourg • France • + 33 (0)9 64 43 69 02

S.A. au capital de 1 011 750 Euros • 316 676 741 RCS EVRY • Code APE 6202A • Siret 316 676 741 000 • TVA Intracommunautaire FR 13316676741

# Summary

<b>Glossary .....</b>	<b>3</b>
<b>1. Introduction.....</b>	<b>4</b>
<b>1 Mini How to. ....</b>	<b>5</b>
1.1 How to process an audio file ? .....	5
<b>2 Main interface .....</b>	<b>6</b>
<b>3 Start a process.....</b>	<b>7</b>
<b>4 Process status and results .....</b>	<b>9</b>
<b>2. MSF model's update interface .....</b>	<b>10</b>
2.1. Word checking .....	10
2.2. Model built.....	13
2.2.1. First step consist in the encoding text checking: .....	14
2.2.2. Second step will display the characters not belonging to a given language :14	
2.2.3. Third step compute the ratio of words which aren't in the dictionary (OOV, out- of-vocabulary) for your input text. ....	14
2.2.4. Fourth step check the encoding of your input dictionary:.....	15
2.2.5. Fifth step check your input dictionary against a list of valid phones:.....	15
2.2.6. Sixth step will merge your dictionary and your current 'master' dictionary:15	
2.2.7. At last the text will be normalized and a last OOV ratio checking is done. 16	
2.3. Model management .....	17
2.3.1. Check models. ....	17
2.3.1. Copy the model. ....	18
2.3.2. Delete the model. ....	18
<b>3. Webservice API.....</b>	<b>19</b>
<b>4. Prononciations setup .....</b>	<b>21</b>
4.1. Common tips.....	21
4.2. Spanish.....	22
4.3. English.....	25
4.4. French.....	28

## Glossary

MS	MediaSpeech, Vecsys speech and language processing system.
Process	Task executed on MS. It means not only transcription but also audio language identification or audio partitioning.
BN	Broadcast News ( TV or radio speech ).
CTS	Conversational Telephone Speech
JobId	The job identifier is a string returned by MS to identify a process.

# 1. Introduction

MSF is a system dedicated for audio transcription. Users can upload audio file onto MSF server, can monitor the transcription process and at last download the transcription result.

This document explains the means for people to process audio file using MSF through its WEB server and how to create a custom model with new words added to the system.

# 1 Mini How to.

## 1.1 How to process an audio file ?

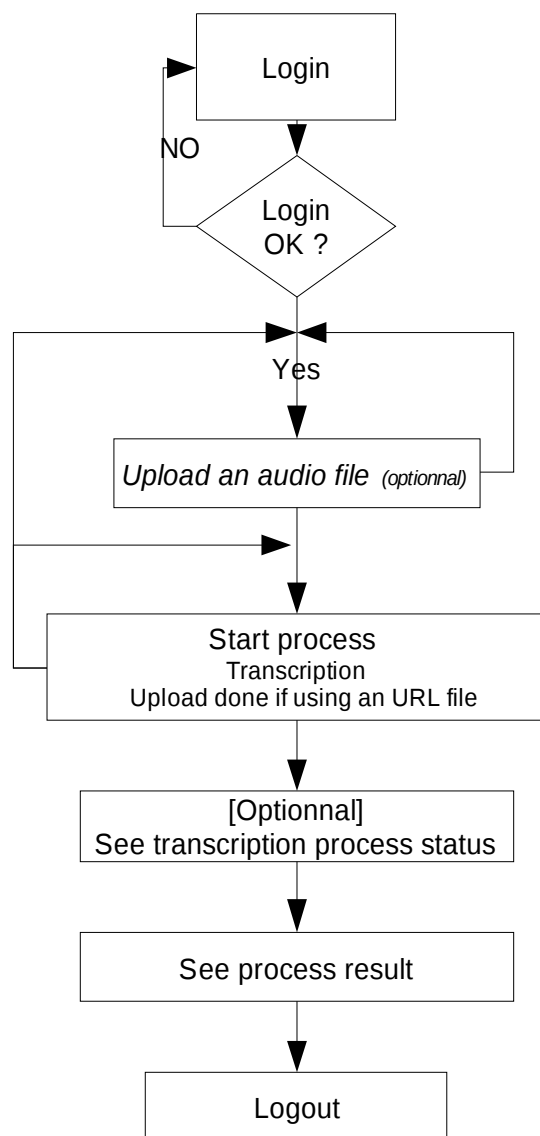
To transcribe the user must first login into MSF, then upload his audio file ( the supported format are the ones readable by the mplayer software ).

Once uploaded, the user can start the transcription process.

The user can upload all its files at once and then start the transcriptions, or upload then start the transcription sequence.

It's possible to follow the progression of the process, and at last see the result of the process.

The user needs to logout, when he stopped accessing MSF.

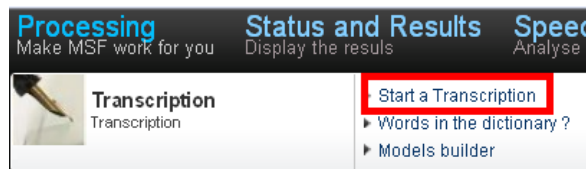


## 2 Main interface



The main menu contains the following items:

The “Processing” menu has a submenu to start a transcription



The “Status and results” menu is used to display the current running processes and view the results



The administration menu is only available for MSF administrator and is neither visible nor accessible to standard users.

### 3 Start a process

The transcription's web page looks like below:

Start the transcription

Mandatory Fields	<input type="checkbox"/> Show Optional fields
<p><b>Audio file type</b> ( an incorrect choice will affect the results )</p> <p>BN - Broadcast news</p> <p><b>Language</b> Français</p> <p><b>Speaker detection</b> Detect speakers (slower)</p> <p><b>Quality of transcription</b> Normal Quality</p> <p><input type="checkbox"/> Use your own language model</p> <p>Model's label : test</p> <p><b>Audio file source</b></p> <p><input checked="" type="radio"/> Upload the audio file from your local computer.</p> <p><input type="radio"/> Choose a web or ftp server, MSF could download from.</p> <p><input type="radio"/> Select some available audio files already on MSF.</p> <p><b>Select some available audio files already on MSF</b></p> <p><input type="checkbox"/> ftpmediasimg#r2d7ftp.vecsys.frtoto.mp4</p> <p><input type="checkbox"/> test.wav.wav</p> <p><input type="checkbox"/> test.wav</p>	

Before starting a transcription you must upload a multimedia file into MSF.

You have two ways to do so:

Upload a file with the web interface:

**1** **Audio file source**

☒ Upload the audio file from your local computer.

☐ Choose a web or ftp server, MSF could download from.

☐ Select some available audio files already on MSF.

**Upload the audio file from your local computer**

**2** Add file...

ads\_err.adt

**3** Upload

Once uploaded the file will appear in the list of files already on MSF.

Attention : If after adding a file to upload the “3” button remains on the state “processing” it means that MSF software has an issue, please contact the technical support to get help, and use FTP to upload your file.

If you want to upload large files, please use a FTP client like filezilla, use the address [mediaspeech.com](http://mediaspeech.com) with your MSF account, put the files in the "media" directory. If you use this transfer mode you must remove all spaces of your file name.

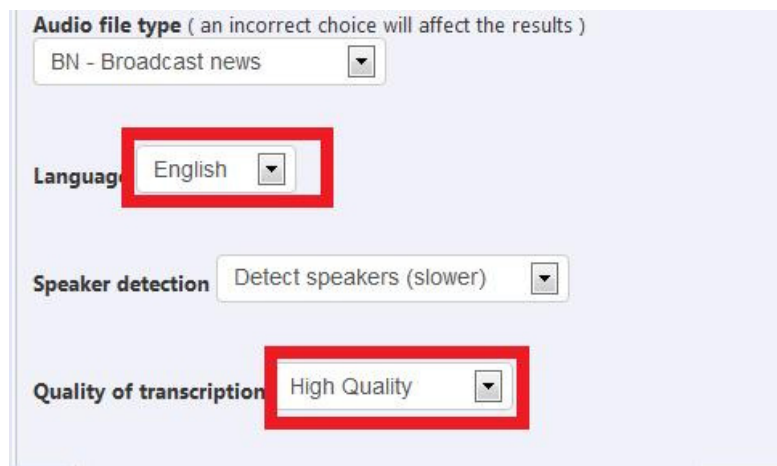
To start a transcription the user has three choices for the audio file source:

The user can upload a file from its computer. He will browse its local data and select an audio file.

The user can make MSF download an audio file from a FTP or Web server.

The user can process a file already on MSF.

To achieve the best quality you must select the following two options:



Audio file type ( an incorrect choice will affect the results )

BN - Broadcast news

Language English

Speaker detection Detect speakers (slower)

Quality of transcription High Quality



## 4 Process status and results

The user can display the status of all its jobs. When a process is completed, links at the right of the table allow the user to view and download the results files.

In this example the results are XML, SRT or Text, but it can also be other kind of formats like LEX.



The screenshot shows the MediaSpeech web interface. At the top, there is a search form with fields for DateStart, DateStop, Job Id, and Audio File Name, along with a User's name dropdown menu. Below the form are buttons for Filter, Clear filter, and Search. The main part of the interface is a table titled "List of processes" with columns for Job Id, Username, Process Name, Src, Hostname, AudioFile, Duration, Status, DateSubmit, DateStart, DateStop, Duration, Ratio, and Results. The table contains four rows of data, each representing a different process.

Job Id	Username	Process Name	Src	Hostname	AudioFile	Duration	Status	DateSubmit	DateStart	DateStop	Duration	Ratio	Results
20325	[?] Play msf	Transcription	WEB	lame01	test-fre.wav	00:00:20	E	2013-09-10 14:40:01	2013-09-10 14:40:01	2013-09-10 14:40:22	00:00:21	1.05	Xml Text Srt Lex traces
19571	[?] Play msf	Alignment	WEB	lame01	test-eng.wav	00:00:20	F	2013-09-02 17:38:21	2013-09-02 17:38:22	2013-09-02 17:38:27	00:00:05	0.25	Xml traces
17407	[?] Play msf	Transcription	WEB	lame01	ra000120130818092000092500000500.wav > general purpose error number	00:05:00	R	2013-08-19 14:18:41	2013-08-19 14:18:42	2013-08-19 14:18:57	00:00:15	0.05	Cancel traces
17406	[?] Play msf	Transcription	WEB	unknown	ra000120130818092000092500000500.wav	00:05:00	F	2013-08-19		2013-08-19			

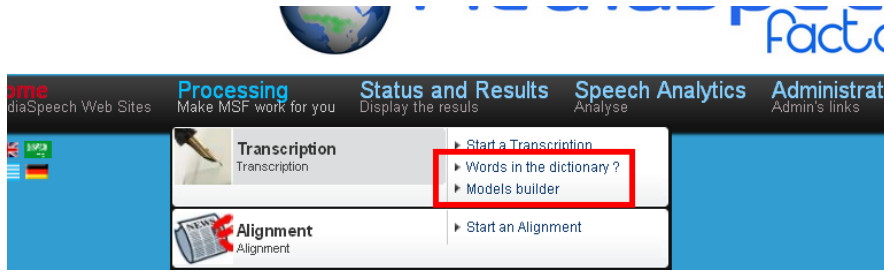
A search function is available, just enter a jobid or a file name and press the Search button.

You can list your process between two dates, for a user level use, the user filter is not active. You can enter a start or/and a stop date then press the button "Filter" to view the results.

The "clear filter" button makes the screen returns to its default values.

## 2. MSF model's update interface

The model menus are located under the transcription menu.



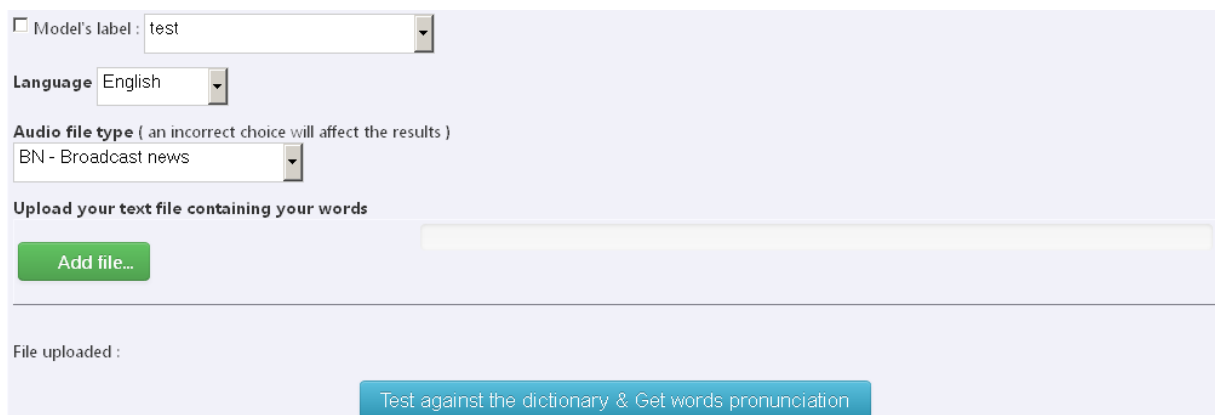
On the displayed page there are three parts.

### 2.1. Word checking

The user can check if a set of words exists in the current dictionary (the default is none specified) or in its current specific model.

To make the test, you must upload a text file; UTF8 encoded, with one word per line, no space.

You have to first add a file, upload it, then press the button "Test..."



Below is an explanation of the result of the vocabulary Query

In order to do a keyword list search against your 'master' dictionary, several query results are displayed in the [log file](#).

First of all, a UTF-8 checking of your keyword list is done:

```
<checkutf8>
> check the UTF-8 encoding of the input and print only the UTF-8 lines to the output
WARNING: not utf8 line : 'Nestlé'
WARNING: line is not processed
> valid lines ~ 99.64% (273/274)
</checkutf8>
```

In this example, one word (i.e. line) is not in UTF-8.

This word will NOT be searched.

If the invalid lines percent rises above a threshold of 1 %, the process exits.

An error will be displayed:

**ERROR: more than 1 % of lines are not UTF-8 encoded**

First query result returned is the list of searched pronunciations which exist in your master dictionary (case sensitive).

First column is the word, all the rest are the corresponding pronunciations.

```
<getinvcs>
Abbey Abi
Abbey's Abiz
Abu abu Abu
Afghanistan AfgAnlstAn afgAnlstAn afganIstan
Afghanistan's AfgAnlstAnz afgAnlstAnz afganIstanz
Airport ERpct ERpct
...
Vuitton vwlt(@|a|O)n v(u|i|w)tan vw?itOn
Waitrose wetroz
Waitrose's wetroz@z
Water wct(R|@)
Water's wct(R|@)z
Waterloo wctRlu wct@lu
Westminster wEstmInstR wEsmInstR wEstmInst@ wEsmInst@
Westminster's wEstmInstRz wEsmInstRz wEstmInst@z wEsmInst@z
William wllj@m wllj@m wlll@m wlll@ms
William's wllj@mz wllj@mz wlll@ms wlll@ms
Wincanton wlnkAnt@n wlnkAntan
Yemen jEm@n
Yemen's jEm@nz
and (A|@)nd (A|@)n And
of @v av Ov
</getinvcs>
```

The second result consists in the words, which are NOT returned previously, finding in case insensitive mode.

The words listed below are always displayed in lower case.

First column is the word, all the rest are the corresponding pronunciations.

```
<getinvci>
asos asoz esoz
astrazeneca Astr@zEnlk@
astrazeneca's Astr@zEnlk@z
coastguard kostgard kosgard
column kOl@m
column's kOl@mz
mastercard mAstRkard mast@kard mast@kad
virus vajrVs
</getinvci>
```

At last, the words which aren't in the dictionary (whatever the case), are returned :

```
<getoovci>
365
A.A
B.M.A
Baroin
...
```

Voser  
W.H.O  
</getoovci>

## 2.2. Model built

A model is built with two kinds of files:

- A dictionary file (.vdic), which contains the list of words and their pronunciations. On each line there is a word and it's pronunciation(s).
- A text corpus file (.txt), which contains a large set of sentences (100 sentences per word) containing the words specified in the dictionary.

The model built and deployments are similar to a transcription's process and listed in the Status and Result web page.

4166581	[?] Play msf	DeployModel	WEB	runner12	na	00:00:00	E	2013-04-03 11:26:07	2013-04-03 11:26:47	2013-04-03 11:27:04	00:00:17	Xml traces
4166493	[?] Play msf	ModelBuilder	WEB	runner12	custometxvdic	00:00:00	F	2013-04-03 11:06:54	2013-04-03 11:06:54	2013-04-03 11:21:38	00:14:44	Xml traces

Before using a new pronunciation or new words, the user must create a new model or overwrite an old one.

Before really building a model, you should test it and check the result contained in the xml result file.

The first step consists in add files to the upload list, and then Starting the upload.

Once uploaded you can test then build your model. The files are deleted from the upload zone when building the model.

To build a model you must upload one vdic file (pronunciations) and one text file (corpus)  
 First Add the files, then Start to Upload.  
 Once uploaded you should write the model's name, select a language and start the process

☐

Model's label :

Language

Audio file type ( an incorrect choice will affect the results )

The already built models are listed in the list box, the button "←"allow copying the name into the Model's label text field.

You must upload 2 files, a text file and a vdic file.

A word means **one** word, you cannot put “me and you” to recognize this set of words. To help recognize such a set of word, you can add some text.

You **MUST** check the result of the built process, below are some explanations to analyze the results. All the result must be read as the result may not be right or wrong, some messages are warning but in some use case may be errors or not even warnings.

Different step of text processing will be applied to your inputs (text, dictionary). Those processing's must be done before building the adapted language model.

For each step, you'll see the displayed **log file**.

### 2.2.1. First step consist in the encoding text checking:

```
<checkutf8>  
> check the UTF-8 encoding of the input and print only the UTF-8 lines to the output  
> valid lines ~ 100.00% (19338/19338)  
</checkutf8>
```

In this example, all lines are encoded in UTF-8.  
Those which aren't, are filtered, and will not be learned during model adaptation.

If the invalid lines percent rises above a threshold of 1 %, the process exits, a error message will be displayed:

**ERROR: more than 1 % of lines are not UTF-8 encoded**

### 2.2.2. Second step will display the characters not belonging to a given language :

```
<checktext>  
> check characters of the input  
WARNING: The origin of the logo may be traced back to a trip made to Łódź city, Poland by  
the 22-year-old Andr Citroen, where he discovered an innovative design for a chevron-shaped  
gear used in milling. , invalid character Ł (hexa code : 0141, name code : LATIN CAPITAL  
LETTER L WITH STROKE)  
WARNING: The origin of the logo may be traced back to a trip made to Łódź city, Poland by  
the 22-year-old Andr Citroen, where he discovered an innovative design for a chevron-shaped  
gear used in milling. , invalid character ź (hexa code : 017A, name code : LATIN SMALL  
LETTER Z WITH ACUTE)  
> valid characters ~ 100.00% (2/2258866)  
</checktext>
```

In this example, two characters are supposed to not be in English.

The words are NOT filtered.

If the invalid character percent rises above a threshold of 1 %, the process exits, a error message will be displayed:

**ERROR: more than 1% of characters are not valid**

### 2.2.3. Third step compute the ratio of words which aren't in the dictionary (OOV, out-of-vocabulary) for your input text.

```
<getoov>
dicutils module ver 1.6
> read the dictionary '/projects/current/msf/vtk/conf/trans/bn/eng/eng-bn.vdic'
> read the vocabulary
> vocabulary vs dictionary '/projects/current/msf/vtk/conf/trans/bn/eng/eng-bn.vdic'
> counts : 427080 words, 0 ccs, 84713 oovs
> lexical scope ~ 21.80% (83137 udic, 45661 uvoc, 27535 uoovs)
> oov ratio ~ 19.84%
> print the out-of-vocabulary
</getoov>
```

If the OOV ratio rises above a threshold of 50 %, the process exits, an error message will be displayed:

**ERROR: oov ratio (50.1%) is greater than threshold (50%)**

#### 2.2.4. Fourth step check the encoding of your input dictionary:

```
<checkutf8>
> check the UTF-8 encoding of the input and print only the UTF-8 lines to the output
> valid lines ~ 100.00% (3061/3061)
</checkutf8>
```

If the invalid lines percent rises above a threshold of 1 %, the process exits, an error message will be displayed:

**ERROR: more than 1 % of lines are not UTF-8 encoded**

#### 2.2.5. Fifth step check your input dictionary against a list of valid phones:

```
<checkaddingdic>
dicutils module ver 1.6
> read the dictionary
> read the dictionary '/projects/current/speech2text/lmadapt/wdir/utf8check.vdic'
> read the list of phones
> read the list of phones from dictionary
WARNING: invalid phone 'F', discard the pronunciation 'glldFAd' from word 'Guildford'
WARNING: invalid phone 'K', discard the pronunciation 'iKotrls@ti' from word 'Ecotricity'
WARNING: invalid phone 'W', discard the pronunciation 'SVt@IWcrT' from word 'Shuttleworth'
WARNING: invalid phone 'W', discard the pronunciation 'SVt@IW@T' from word 'Shuttleworth'
WARNING: invalid phone 'K', discard the pronunciation 'iKov@' from word 'Ecover'
> write output dictionary file
</checkaddingdic>
```

Invalid pronunciations are discarded.

If at least one word has no more pronunciations, the process exits, an error message will be displayed:

**ERROR: 1 word are rejected from the input dictionary**

#### 2.2.6. Sixth step will merge your dictionary and your current 'master' dictionary:

```
<mergedic>
added 3M TrijEm TriEm
...
update Guildford glldfRd gllfRd gllfAd gllfAd -> glldfRd gllfRd gllfAd
...
updates: 172
added: 192
</mergedic>
```

This process will display the changes done against your current 'master' dictionary ( "words added" and "pronunciation updated" ).

### 2.2.7. At last the text will be normalized and a last OOV ratio checking is done.

```
<normtext>
dicutils module ver 1.6
> read the dictionary '/projects/current/speech2text/lmadapt/wdir/add2dic.vdic'
ignore duplicated phonetic 'al' for word 'Al'
ignore duplicated phonetic 'kApElo' for word 'Capello'
ignore duplicated phonetic 'dZemz' for word 'James'
ignore duplicated phonetic 'And' for word 'and'
ignore duplicated phonetic 'At' for word 'at'
> print the vocabulary from dictionary
> get exception word list '/projects/current/speech2text/lmadapt/wdir/add2dic.voc'
> get text from input
</normtext>

<getoov>
dicutils module ver 1.6
> read the dictionary '/projects/current/speech2text/lmadapt/wdir/eng-bn.vdic'
> read the vocabulary
> vocabulary vs dictionary '/projects/current/speech2text/lmadapt/wdir/eng-bn.vdic'
> counts : 450529 words, 0 ccs, 27694 oovs
> lexical scope ~ 24.25% (83329 udic, 29260 uvoc, 9051 uoovs)
> oov ratio ~ 6.15%
> print the out-of-vocabulary
</getoov>
```

If the OOV ratio, for this last step, rises above a threshold of 20 %, the process exits:

**ERROR: oov ratio (20.1%) is greater than threshold (20%)**

If one of those steps failed (exit), then the model language cannot be build before the inputs aren't modified.



## 2.3. Model management

This part is the most important and risky.

You can brick your transcription if you make some mistake.

**You should not have more than two models at the same time on the blades.**

A check will be developed on MSF to avoid such a situation.

Be aware that:

- MSF always let you copy a model on the blades unless the free disk space reaches 80%.
- If you deploy an invalid model your transcriptions on MSF won't work anymore.
- You cannot test a model before deploying it.

Three functions are available to help you manage you models.

This operation consists in checking the model of all blades (this can take a little time ...)  
Do not refresh your browser until it's completed  
Check models

**Be very carefull : This operation will copy the model generated above.  
Before pressing the below button you MUST have checked the result of the model built.  
If you copy an invalid model, the transcription will NOT work anymore for your account !**

Model's label :   
Copy the customer transcription's system into the Blades

**Be very carefull : This operation will DELETE the selected model above.**

Model's label :   
Delete the customer's model from the Blades

### 2.3.1. Check models.

This function accesses the frontend and the blades and lists all the available models.

The “.vdic” file is your dictionary, the “.dmp” file is your language’s model built with your text file.

You can check the dates and the file sizes.

Checking models ...

Frontend!! (here are located your master models) !! :

- Apr 3 11:20 : model1/eng-bn.tg.dmp (754101000 bytes)
- Apr 3 11:21 : model1/eng-bn.vdic (1483887 bytes)

runner12 :

- Apr 3 11:21 : model1.20130403/eng-bn.vdic (1483887 bytes)
- Apr 3 11:20 : model1.20130403/eng-bn.tg.dmp (754101000 bytes)
- Apr 3 11:21 : model1/eng-bn.vdic (1483887 bytes)
- Apr 3 11:20 : model1/eng-bn.tg.dmp (754101000 bytes)

### 2.3.1. Copy the model.

This function make the copy of your model from the frontend into the blades, this process is listed in your list of processes as "DeployModel".

You should check the result of the copy.

```
Deployment Starting ...  
Copy model into blade runner12 done, code=0  
Deployment completed, code=0
```

### 2.3.2. Delete the model.

This function delete your model from the frontend and the blades, this process is listed in your list of processes as "DeployModel" just like the copy, you have to check the xml to know if the action was a copy or a deletion.

You should check the result of the deletion.

```
Deployment Starting ...  
Deleting model /Models/1/backup.20130403 on blade runner12 done, code=0  
Deployment completed, code=0
```

### 3. Webservice API

The model's name is named "corpus" in the webservice API.

The below specification is extracted from the webservice API document.

Allow a user to start the transcription of an audio file previously uploaded or a WEB located file.

The function name is:

**trans**

Arguments:

<b>filename</b>	String	File name of the audio to process
<b>type</b>	String Values: cts bn	Specify the type of audio file. CTS: Telephonic conversation. <i>(if component present)</i> BN: Broadcast. <i>(if component present)</i>
<b>language</b>	String Values: fre eng ara chi spa rus	Specify the language of the audio file. <b>Mandatory with BN audio file.</b> <i>If not specified MS will automatically identify the language (if optional component present).</i>
<b>channel</b>	integer Values: $\frac{1}{2}$	Specify the channel to decode. <i>If the audio file is stereo, specify the channel 1 or 2.</i> <b>ONLY with CTS audio file.</b>
<b>quality</b>	Integer Values: <u>0</u> 1 2	used to parameter the transcription <i>for MSF &gt; v2.40</i> 0: normal quality. 1: medium quality. 2 : high quality
<b>corpus</b>	<b>string</b>	<b>Name of the user's model, empty if none.</b>
<b>duration</b>	Integer	Length of the sound track to process ( s ).
<b>callback</b>	String	Url called when the status of a job change.
<b>gpgkey</b>	String	GPG password.

Result:

The result returned by this function is the standard result format plus additional information, the process identifier ( JobId). **The Job id is a string with digits.**

The result of the transcription start process is displayed in the jobid field.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE result SYSTEM "MsfResults.dtd">
<result>
<status>Error Code</status>
<message>Error Message</message>
<jobid>xxxx</jobid>
</result>
```

## 4. Pronunciations setup

This chapter will help you to create the file containing the pronunciation of your words.

The phones used are specific for each language.

### 4.1. Common tips

The same word can have various pronunciations. So, variants are listed one after another.

In order to avoid a long list and a difficult proofreading, it is possible to use a compact phonetization which represents with a single phonetization several pronunciations.

This is possible thanks to some patterns:

- The quantifier "?" follows a phoneme which can be pronounced or not  
Example : the French word 'petit' is pronounced either « pxti » or « pti ». The compact phonetization is « px?ti ».
- The operator « | », enclosed in parenthesis, means « or ». Thus, phonemes in parenthesis and separated by « | » are the various alternatives.  
Example : the word 'agenda' is pronounced in French either « aZAda » or « aZlda ». The compact phonetization is « aZ(A|l)da ».
- A same word can combine different patterns.  
Example : the word 'email', in French, is represented by « im(e|E)j ?l ». This phonetization corresponds to the following pronunciations : « imEl », « imEjl », « imel » et « imejl »

-NOTE: The compact phonetization can generate some errors like « (r|Z)o(s|z)e » to the word 'José'. This phonetization represents the correct pronunciations « rose » and « Zoze » but it also represents the incorrect pronunciations « roze » et « Zose »

In this case, it is better to list the various phonetizations

## 4.2. Spanish

Lista de fonemas utilizados para el español:

Vocales átonas:

Fonemas	Correspondencia ortográfica
a	cas[a]
e	c[e]reza
i	p[i]scina
o	c[o]rona
u	ac[u]mulo

Vocales tónicas:

Fonemas	Correspondencia ortográfica
A	c[a]sa, c[á]mara
E	cer[e]za, c[é]sped
I	pisc[i]na, l[i]mite
O	cor[o]na, [ó]rdago
U	acum[u]lo, n[ú]mero

Semiconsonantes:

Fonemas	Correspondencia ortográfica
j	h[i]elo
w	h[u]eco

Consonantes:

Fonemas	Correspondencia ortográfica
b	[b]eso, pol[v]o
d	[d]inero
f	[f]eliz
g	[g]ato
k	[c]arne, [k]ilo, [qu]eso
l	[l]oro
L	[ll]oro, a[y]untamiento
m	[m]ano
n	[n]unca
N	a[ñ]orar
p	[p]olvo
r	pe[r]o
R	pe[rr]o, [r]omano

s	[s]olar
S	[sh]ow
t	[t]iempo
T	[ch]ocolate
x	[j]inete, [g]enio
z	[c]epillo, [z]apato

Ejemplos de fonetización:

casa	kAsa
cereza	zerEza
piscina	piszIna
corona	korOna
acumulo	akumUlo
cámara	kAmara
césped	zEsped
límite	llmite
órdago	Ordago
número	nUmero
hielo	jElo
hueco	wEko
beso	bEso
dinero	dinEro
feliz	fellz
gato	gAto
carne	kArne
kilo	kIlo
queso	kEso
loro	lOro
lloro	LOro
ayuntamiento	aLuntamjEnto
mano	mAno
nunca	nUnka
añorar	aNorAr
polvo	pOlbo
pero	pEro
perro	pERo
romano	RomAno
solar	solAr

show	Sou
tiempo	tjEmpo
chocolate	TokolAte
jinete	xinEte
genio	xEnjo
cepillo	zepILo
zapato	zapAto



## 4.3. English

English Vecsys Phone Set:

Long Vowels:

Phonemes	Orthographic fulfillment
a	f[a]ther
i	sh[ee]p, [ea]rs
c	[a]ll, h[o]rse, l[aw], c[au]ght
u	r[u]le, p[oo]l, l[o]se

Short Vowels:

Phonemes	Orthographic fulfillment
A	th[a]t
E	b[e]d, h[ea]d
I	b[i]g
O	l[o]t
U	f[oo]t, p[u]t
V	b[u]t, gl[o]ve, en[ou]gh
@	[a]bove

Diphthongs and semivowels:

Phonemes	Orthographic fulfillment
e	d[a]te
o	n[o]se, c[oa]t, sh[ow]
j	[y]es, [u]nion, co[i]n
w	[w]e, mo[u]th

Consonants:

Phonemes	Orthographic fulfillment
b	[b]ook
d	[d]ay
D	[th]e
dZ	[j]ump
f	[f]ish
g	[g]o
h	[h]and
k	ta[k]e, pa[ck], [c]old
l	[l]ook
m	[m]other

n	[n]o
N	si[ng]
p	[p]en
r	[r]ed
R	bi[r]d
s	[s]ay, tra[c]e
S	[sh]e
t	[t]own
T	[th]ink
tS	wa[tch]
v	[v]ery
z	[z]one, rai[s]e
Z	vi[s]ion

Examples:

Word	Phonetization
father	faDR
sheep	Sip
ears	irz
all	cl
horse	hcrs
rule	rul
pool	pul
that	DAt, D@t
bed	bEd
head	hEd
big	blg
lot	lOt
foot	fUt
but	bVt
glove	glVv
above	@bVv
date	det
nose	noz
coat	kot
show	So
yes	jEs
union	junj@n

we	wi
mouth	mawT
book	bUk
day	de
the	D@, Di
jump	dZVmp
fish	fIS
go	go
hand	hAnd
take	tek
pack	pAk
cold	kold
look	lUk
mother	mVDR
no	no
sing	sIN
pen	pEn
red	rEd
bird	bRd
say	se
trace	tres
she	Si
town	tawn
think	TINk
watch	watS
very	vEri
zone	zon
raise	rez
vision	viZ@n
eye	aj
now	naw
coin	kcjn
law	lc
caught	kct
put	pUt
enough	EnVf, inVf
lose	luz

## 4.4. French

Jeux de phonèmes pour le français :

Voyelles :

Phonèmes	Orthographe correspondante
a	p[a]tte, p[â]te
e	cl[e]f, cl[é]
E	g[e]l, cr[è]me, [ai]mer, m[ai]tre, f[ê]te
@	p[e]tit, c[œu]r, l[eu]r
Y	f[eu], c[eux]
i	l[i]t, [y], [î]le
c	s[o]l
o	s[au]le, h[ô]tel, h[au]t, bur[eau]
u	f[ou]
y	s[u]c, s[û]r

Voyelles nasales :

Phonèmes	Orthographe correspondante
A	ch[an]t, v[en]t
I	br[in], br[un], chi[en], tr[ain], pl[ein]
O	b[on]

Semi-voyelles :

Phonèmes	Orthographe correspondante
j	[y]ole, fi[l]le, trava[i]l, f[i]ère
w	[ou]i, m[o]yen, [w]eb
h	l[u]i

Consonnes :

Phonèmes	Orthographe correspondante
b	[b]on
d	[d]on
f	[f]ou, [ph]armacie
g	[g]ond, [gu]êpe
k	[c]ou, [k]ilo, ar[ch]ai[qu]e
l	[l]ong
m	[m]ont, fe[mm]e
n	[n]om, bo[nn]e
N	di[gn]e

p	[p]ond
r	[r]ond, [rh]ume
s	[s]ot, [ç]a, chau[ss]ure, cau[t]ion, [c]e, [sc]ie
S	[ch]at, [sh]ampooing, [sch]éma
t	[t]on, [th]é
v	[v]in, [w]agon
z	[z]èbre, rai[s]in
Z	[j]our, [g]îte

Exemples :

Mot	Phonétisation
patte	pat, pat@
pâte	pat, pat@
clef	Kle
clé	Kle
gel	ZEI
crème	krEm, krEm@
aimer	Eme
maître	mEtr
fête	fEt
petit	p@ti
cœur	kWr
feu	fY
ceux	cY
lit	Li
y	I
île	Il
sol	scl
saule	sol
hôtel	otEl
haut	O
bureau	byro
fou	Fu
suc	syk
sûr	syr
chant	SA
vent	vA

brin	brl
brun	brl
chien	Sjl
train	trl
plein	pll
bon	bO
yole	jcl
filie	fij
travail	travaj
fière	fjEr
oui	wi
moyen	mwajl
web	wEb
lui	lhi
don	dO
pharmacie	farmasi
gond	gO
guêpe	gEp
cou	ku
kilo	kilo
archaïque	arkaik
long	lO
mont	mO
femme	fam
nom	nO
digne	diN
bonne	bcn
pond	pO
rond	rO
rhume	rym
sot	so
ça	sa
chaussure	Sosyr
caution	kosjO
ce	s@
scie	si
chat	Sa
shampooing	SApwl
schéma	Sema

ton	tO
thé	te
vin	vl
wagon	vagO
zèbre	zEbr
raisin	rEzl
jour	Zur
gîte	Zit
leur	lWr