CREATING A SPEECHWEB APPLICATION

SpeechWeb Developers Manual Version 03/08

C Version

This manual has been written to help developers create a SpeechWeb application (in C programming language in this version)

24/3/2008

INTRODUCTION

The Public-Domain SpeechWeb is a web of speech accessible applications created by faculty and students at the School of Computer Science at the University of Windsor. The SpeechWeb is based on a novel architecture invented by Dr. Richard Frost, and described in the paper "Call for a Public-Domain SpeechWeb" Commun. ACM 48(11): 45-49 (2005)

http://davinci.newcs.uwindsor.ca/~richard/speechweb_papers/CACM_SpeechWeb.pdf

This manual explains how to create and deploy your own speech applications.

You should begin by installing the Opera browser, downloading the IBM speech recognition plugin for Opera, and then using Opera to access an existing SpeechWeb application. The instructions on how to do this are given in the following paper: Richard A. Frost, Xiaoli Ma, Yue ShiA browser for a public-domain SpeechWeb. WWW 2007: 1307-1308

http://davinci.newcs.uwindsor.ca/~richard/speechweb_papers/WWW_2007.pdf

After experimenting with existing SpeechWeb applications, you are ready to create your own application.

CREATING A SPEECHWEB APPLICATION
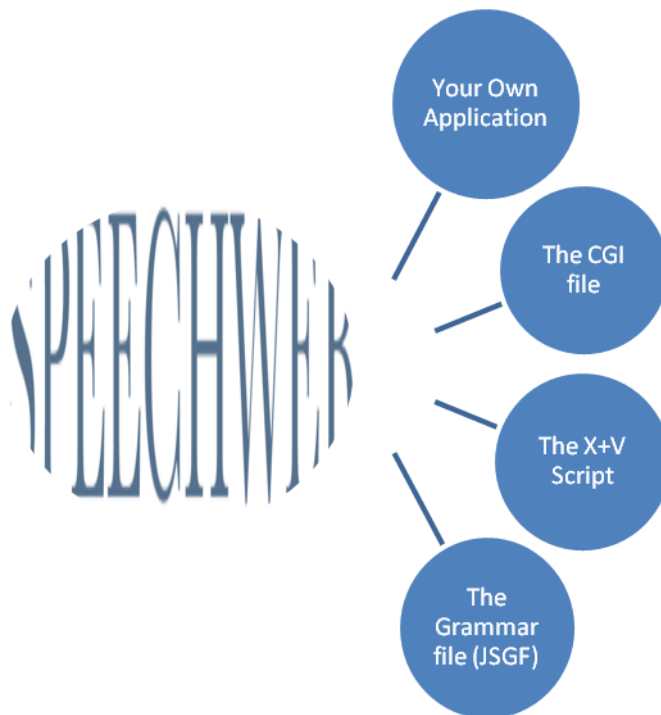
Instructions

We assume that you will use a Unix server for your application, which will be written in C language. We will show later how to use other servers and programming languages. (A Miranda version of this manual can be found on the SpeechWeb website)

A) Create a directory called "public_html" in your home directory on the School of Computer Science Solaris server (or another server which supports the CGI web communications protocol). This how to create the directory:

Open Files manager > Click on file > New folder > rename it to public_html

To make the public_html directory accessible by others:

 Right click on public_html > properties > Change the Permission mode to 755,

 Or select the following after clicking on the public_html directory icon:

B) Inside the public_html directory, you need to create a new directory and rename it with the name of your application. For example, in this manual, we shall refer to this directory as "name_of_your_application".

> You need to change the Permission mode to 755 for this directory.

C) Inside "name_of_your_application", you will need to add four files. To make this easy, you can copy four files from the existing application "andros" and then modify them for your own application as follows:

1. The cgi executable

Go to the following link:

http://luna.cs.uwindsor.ca/~karabi/andros/andros.demo

You will get the .cgi file for andros's SpeechWeb application:

```
#!/bin/csh -f

setenv HOME '/stu2/karabi/public_html/andros'

echo "Content-Type:text/plain"
echo ''

setenv v "`/bin/cat`"
/stu2/karabi/public_html/andros/andros << zzz
$v
zzz
```

You need to copy the contents of this file into a text editor, save it in the directory "name_of_your_application" and rename it as name_of_your_application.cgi (Make sure that the contents of the page were copied into the text editor in the same order they are and that there was no new characters added to the file such as a new line. It is best to copy this file using the Unix system rather than through a PC which can add new characters))

Now, after you have created your own CGI file, you need to change some of the contents to make it work with your own application:

On line 3, change: '/stu2/karabi/public_html/andros'

To: '/stu<stu_num>/<your_user_name>/public_html/name_of_your_application'

<stu_num> is the number of the stu directory your account is in. You can find it on the top of the file manager:

In this example, it is stu2.

<your_user_name> is your Uwindsor user name you use to log on to a sun terminal.

<name_of_your_application> is the directory we have created in part B).

For example, the full path name would be something like the following:

'/stu2/karabi/public_html/name_of_your_application'

On line 10 change:

/stu2/karabi/public_html/andros/andros << zzz

 to:

/stu<stu_num>/<your_user_name>/public_html/name_of_your_application
/name_of_your_executable>

<name_of_your_executable> will be the name of the executable "a.out" generated by compiling your C program, you can rename the executable "a.out" to the same name you gave to your application.

Now, save the .cgi file and close it.

Finally, change the Permission mode to 755 for the .cgi file.

Open the link: http://cs.uwindsor.ca/~karabi/andros/andros.xml

If you are using a browser other than Opera you will get an error page:

XML Parsing Error: undefined entity
Location: http://luna.cs.uwindsor.ca/~karabi/andros/andros.xml
Line Number 380, Column 49:

```
objTable.rows[0].cells[0].innerHTML = "<br/> ";
-------------------------------------------------^
```

That's fine, because you only need the source code of the page.

Just copy the source code into a text editor, save it to "name_of_your_application" and rename it as name_of_your_application.xml.

This is how to find the source code:
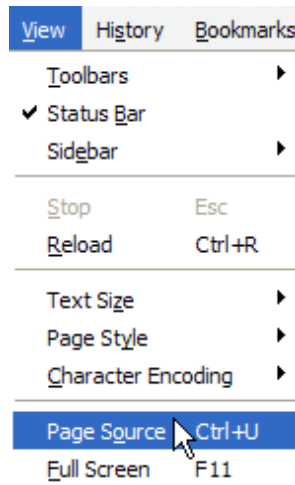
There are two main changes you need to make in the xml file:

1. On line 20:

Change: var sv_greeting ="hi, my name is Andros. If you need help, please say help at any time.";

To: var sv_greeting ="hi, my name is <name_of_your_application.";

This greeting message will be said to the user only at the first time the user visits this page. So you are free to put the greeting you want, only keep it short.

2. On line 23:

Change: http://luna.cs.uwindsor.ca/~karabi/andros/andros.cgi";

To: "http://cs.uwindsor.ca/~your_user_name/name_of_your_application/ name_of_your_application.cgi>";

<name_of_your_application.cgi> is the name you gave to your cgi file.

3. On line 31:

Change: "http://luna.cs.uwindsor.ca/~karabi/andros/andros.jsgf"

To: "http://cs.uwindsor.ca/~your_user_name/name_of_your_application/ name_of_your_application.jsgf"

<name_of_your_application.jsgf> is the name of your grammar file.

Now, save the .xml file and close it.

Finally, change the Permission mode to 755 for the .xml file.

3. The grammar file:

You need to create a grammar file and add it to "name_of_your_application". The name of your grammar file must be name_of_your_application.jsgf.

JSGF is the Java Speech Grammar Format which is a language that you can use in order to write an attribute grammar for your application. (Refer to "Useful links" to see some examples).

4. Your application:

Add your application to "name_of_your_application" and make the following change:

The CGI communication protocol will add 9 characters to the input to the speechweb browser, before sending it to your application. In order to get your application to work properly, you need to drop these 9 characters from the input. Below is an example of a function that drops the 9 characters from a String:

```
/***drop9: drops the first 9 characters of S.
requires #include<string.h> ***/

void drop9( char *S );

void drop9( char *S )
{
        char temp[501];

        strcpy( temp, S );
        strcpy( S, temp+9 );
}
```

D) Useful links:

- ✓ The SpeechWeb website:
  http://sol.cs.uwindsor.ca/~speechweb/p_d_speechweb/index.html
- ✓ Attribute grammars written using JSGF:

  - ➢ Simple grammar:
    http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/judy/judy.jsgf

  - ➢ More complex grammar:
    http://luna.cs.uwindsor.ca/~karabi/ask/ask.jsgf
    http://luna.cs.uwindsor.ca/~karabi/andros/andros.jsgf
    http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/solarman/solar
    man.jsgf

- ✓ Application written using C language:
  http://luna.cs.uwindsor.ca/~karabi/andros/andros.c

- ✓ Grammar Format Specification (JSGF):
  http://java.sun.com/products/java-
  media/speech/forDevelopers/JSGF/JSGF.html#11803

- ✓ Any suggestions can be forwarded to: karabi@uwindsor.ca

E) To Test your application.

Now use Opera with the IBM speech recognition plugin to access your application by directing
Opera to the following URL:

www.cs.uwindsor.ca/your_user_id/name_of_your_application/name_of_your_application.cgi