

Using Nuance® UniMRCP Sample Client 1.0

Nuance® UniMRCP Sample Client 1.0 is an MRCPv2 client implementation using the UniMRCP framework. It exercises basic functionality of a Nuance speech-processing system. You can use the source code, which is included with the installation package, as a basis for your own MRCP client.

UniMRCP Sample Client runs multiple iterations of multiple applications, based on command-line settings. Each iteration for each application runs the following sequence of events:

- 1 Generate a random number.
- 2 Speak the number sequence as digits (e.g., “one , two, four”) and save the audio to a file.
- 3 Speak the number as a whole (“one hundred and twenty four”) and save the audio to a file.
- 4 Recognize the digits in the number from the audio file saved in step 2 and save the recognition result in a file.
- 5 Recognize the number as a whole from the audio file saved in step 3 and save the recognition result in a file.

Supported environments

The client has been tested on the following operating systems and against the following Speech Server versions:

Operating systems

- Centos 5.4 64-bit
- Windows XP 32-bit
- Windows 2008 R2 (64-bit)

Note: UniMRCP Sample Client is built as a 32-bit or 64-bit application for the corresponding system architectures.

Speech Server versions

- NSS 5.1 and NSS 6
- Local and remote NSS (same and different host)

UniMRCP Sample Client

UniMRCP Sample Client consists of the file *nuance-unimrcp-sample-client-1.0.zip*, which you download from Nuance Network, plus packages distributed by UniMRCP (<http://www.unimrcp.org/>).

UniMRCP Sample Client packages

The file *nuance-unimrcp-sample-client-1.0.zip* contains the following folders and files:

- **data**: destination folder for audio files from SPEAK requests and recognition results from RECOGNITION requests (the following sample files are provided).

synth-digits-3964.pcm: audio received as a result of a SPEAK request for individual digits
synth-number-3964.pcm: audio received as a result of a SPEAK request for a whole number
recog-digits-3964.txt: result received as a result of a RECOGNIZE "digits" request
recog-number-3964.txt: result received as a result of a RECOGNIZE "a number" request

- **log**: execution log if turned on (see [Configuring UniMRCP Sample Client on page 5](#) for more details).

unimrcp_example.log: example of a log from a test execution.

- **src**

unimrcp-sample-client.c: all source code and declarations for the UniMRCP Sample Client (using tab size 2)
Makefile: for building under Linux
Release Notes.pdf: this file

- **include**: contains APR header files. (You must copy them to the APR include directory; see [Installing and compiling UniMRCP Sample Client on page 3](#).)

apr_xml.h
apu.h

- **conf**

unimrcpclient.xml: configuration file (references nuance.xml)
client-profiles/nuance.xml: configuration for connecting with Nuance Speech Server

UniMRCP packages

UniMRCP Sample Client has the following dependencies:

- UniMRCP 1.0.0
- APR 1.4.2
- APR-Util 1.3.9
- Sofia-SIP 1.12.10

All required dependencies are available in the following packages distributed by UniMRCP (<http://www.unimrcp.org/>):

Linux

- *unimrcp-deps-1.1.0.tar.gz*
- *unimrcp-1.0.0.tar.gz*

Windows

- *unimrcp-sdk-1.0.0.exe* and *unimrcp-1.0.0.exe* (32-bit)
- OR
- *unimrcp-x64-sdk-1.0.0.exe* and *unimrcp-x64-1.0.0.exe* (64-bit)

Installing and compiling UniMRCP Sample Client

Linux

- 1 Download and install the installation packages from <http://www.unimrcp.org/>:

```
unimrcp-deps-1.1.0.tar.gz
unimrcp-1.0.0.tar.gz
```

- 2 Download the file *nuance-unimrcp-sample-client-1.0.zip* from Nuance Network.

- 3 Build UniMRCP Sample Client:

- a Unzip *nuance-unimrcp-sample-client-1.0.zip* to any location.
- b Add files *apr_xml.h* and *apu.h* provided under *../nuance-unimrcp-sample-client-1.0/include* to the apr include folder. The default location is */usr/local/apr/include/apr-1/*.

Note: These files will be installed as part of UniMRCP in a future release.

- c Add apr library directory to LD_LIBRARY_PATH. The default location is */usr/local/apr/lib*.

- d Change directory:

```
cd ../nuance-unimrcp-sample-client-1.0/src
```

- e Run make clean (not required the first time).

- f Run make.

- g Run *./unimrcp-sample-client* [options].

Notes:

- Cross compiling was not tested; that is, UniMRCP Sample Client and the UniMRCP libraries were compiled according to the operating system architecture (32 vs 64 bit).
- Make sure *zlib-devel.x86_64* & *zlib-devel.i386* are installed; otherwise, you will get the following error when installing SOFIA:

```
> configure: error: "printf cannot handle 64-bit integers"
```

- Adjust the makefile if:
 - UniMRCP components are not installed in their default location.
 - You want to compile using 32 bit.

Adjust the following parameters in such cases:

```
UNIMRCP_HOME = /usr/local/unimrcp
APR_HOME     = /usr/local/apr/
TARG_ARCH   = -m64
```

Windows

- 1 Download and install the installation packages from <http://www.unimrcp.org/>:

- *unimrcp-sdk-1.0.0.exe* and *unimrcp-sdk-1.0.0.exe* (32-bit)

OR

- *unimrcp-x64-sdk-1.0.0.exe* and *unimrcp-x64-1.0.0.exe* (64-bit)

- 2 Download the file *nuance-unimrcp-sample-client-1.0.zip* from Nuance Network.

3 Set up your building environment:

- a Unzip *nuance-unimrcp-sample-client-1.0.zip* to a location accessible and convenient for your build setup.
- b Include directory: *<UniMRCP home>\include*.
- c Add files *apr_xml.h* and *apu.h* provided under *..\nuance-unimrcp-sample-client-1.0\include* to the UniMRCP include folder *<UniMRCP home>\include*. The default location is *C:\Program Files\UniMRCP\include*.
Note: These files will be installed as part of UniMRCP in a future release.
- d Preprocessor definitions required: WIN32, APT_STATIC_LIB, MPF_STATIC_LIB & MRCP_STATIC_LIB.
- e Additional library directories: *<UniMRCP home>\lib*.
- f Dependencies=

```
mrcp.lib  
mpf.lib  
aprtoolkit.lib  
libaprutil-1.lib  
libapr-1.lib  
libunimrcpclient.lib  
mrcpsofiasip.lib  
mrcpunirtsp.lib  
unirtsp.lib  
mrcpclient.lib  
mrcpv2transport.lib  
mrcpsignaling.lib  
libsofia_sip_ua.lib  
ws2_32.lib  
winmm.lib
```

- g Add source file *..\unimrcp-sample-client\src\unimrcp-sample-client.c* in the project.

4 Compile UniMRCP Sample Client as a console application.

5 Run UniMRCP Sample Client.

Notes:

- Cross compiling was not tested; that is, UniMRCP Sample Client and the UniMRCP libraries were compiled according to the operating system architecture (32 vs 64 bit).
- When running UniMRCP Sample Client, directories "conf", "data" and "log" are accessed according to a base directory value.
- The default location for the base directory is "..\", so the executable must be in a sibling folder of directories "conf", "data", and "log"; or,

The command line argument *-r <base path value>* must be provided to point to the parent directory of directories "conf", "data" and "log".

Configuring Nuance Speech Server

Number of licenses

The license settings must support the number of RECOGNIZE / SPEAK requests UniMRCP Sample Client will issue simultaneously. In essence, the number of Recognizer and Vocalizer license ports and the maximum number of allowed SIP sessions must be greater or equal to the number of applications (command line argument 'a').

For example, if UniMRCP Sample Client runs 500 applications (*-a 500*), set the following parameters:

\$NSSVRSDK/config/NSSserver.cfg

```
server.mrcp2.sip.maxCountOfSession          VXIInteger      500
```

\$SWISRSDK/config/Baseline.xml

```
<!-- How many osr_swirec licenses will be checked out during SWIrecInit() -->
<param name="swirec_license_ports">
  <value>500</value>
</param>

<!-- How many osr_swiep licenses will be checked out during SWIepInit() -->
<param name="swiep_license_ports">
  <value>500</value>
</param>
```

\$VOCALIZER_FOR_NETWORK/conf/baseline.xml (*ttsrshclient.xml* for NSS 5.1)

```
<tts_license_ports>500</tts_license_ports>
<tts_license_ports_overdraft_thresh>0</tts_license_ports_overdraft_thresh>
<cpr_license_ports>500</cpr_license_ports>
<cpr_license_ports_overdraft_thresh>0</cpr_license_ports_overdraft_thresh>
```

SSML validation

SSML strict mode must be set to 'none':

```
<ssml_validation>none</ssml_validation>
```

Configuring UniMRCP Sample Client

In the configuration file *../conf/client-profiles/nuance.xml* set the following parameters:

```
"...
<settings>
  <!-- SIP MRCPv2 settings -->
  <sip-settings id="Nuance-SIP-Settings">
    <!-- Server address should be explicitly specified, it defaults to "ip" address set in
         the properties. -->
    <server-ip>127.0.0.1</server-ip>
    <server-port>5060</server-port>
    <!-- <force-destination>true</force-destination> -->
  </sip-settings>
..."
```

Notes:

- If the client is on the same host as the server, set the server IP address to 127.0.0.1.
- The port must be set to the value of parameter *server.mrcp2.sip.transport.tcp.port* in *\$NSSVRSDK/config/NSSserver.cfg*. The default value is 5060.

Running UniMRCP Sample Client

The client supports the following options:

Option	Description	Default
-a	Number of applications, or execution threads, running in parallel.	1
-i	Number of iterations per execution thread.	1
-o	Log output (0-none, 1-console only, 2-file only, 3-both).	3
-r	Base path of directories 'conf', 'log' and 'data' used at runtime.	../

Notes:

- Execution steps are shown on the console and saved in the log file *../log/unimrcpclient-0.log* (controlled by the command line argument -o).
- For each SPEAK request, the audio received from the MRCP server is saved in a file under folder *../data*.
- For each RECOGNITION request, the result is saved in a file under folder *../data*.

Example:

```
./unimrcp_sample_client -a 2 -i 2
```

This would result in two applications, or execution threads, performing two iterations each. This should generate 4 “numbers” and 16 files.

For example, if the numbers generated at runtime are 41, 3599, 6789 and 7669, we would have the following files under *../data*:

```
synth-digits-41.pcm
synth-number-41.pcm
recog-digits-41.txt
recog-number-41.txt
synth-digits-3599.pcm
synth-number-3599.pcm
recog-digits-3599.txt
recog-number-3599.txt
synth-digits-6790.pcm
...same pattern continues...
recog-number-7669.txt
```