Powered by Universal Speech Solutions LLC

Server Integration Manual

Developer Guide

Revision: 41

Last Updated: May 20, 2017

Created by: Arsen Chaloyan

Table of Contents

1 Overview

This guide describes how to integrate the UniMRCP server library into a user application. The document provides basic steps only and is not a complete reference.
The intended audience is speech application developers familiar with the C/C++ programming languages.

1.1 Applicable Versions

Unless explicitly stated, instructions provided in this guide are applicable to the following versions.

UniMRCP 1.0.0 and above

1.2 Installation and Configuration

For the installation and configuration of the library, see the Installation Guide and the Server Configuration Guide.

2 Architecture

The UniMRCP server library provides a server-side implementation of the MRCP v1 and v2 protocols. The UniMRCP server stack is a multi-threaded library implemented in the C language and built on top of a number of internal and external components. The library uses the asynchronous event-driven model with a fixed number of threads (tasks) launched upon initialization. Memory pools are used throughout the library in order to avoid dynamic memory allocations and, thus, make memory operations highly efficient.

3 Server Stack Initialization

Typically, one instance of the server stack is created per running process. The server stack is loaded from the configuration file(s). The location of the configuration directory is specified by the structure apt_dir_layout_t.

```
/* create default directory layout relative to root directory path */
apt_dir_layout_t *dir_layout =
apt_default_dir_layout_create(root_dir_path,pool);
/* start server stack processing */
mrcp_server_t *server = unimrcp_server_start(dir_layout);
```

4 Server Stack De-initialization

```
/* shutdown server stack processing */
unimrcp_server_shutdown(server);
```

This function stops the message processing loop, terminates all other activities in the server stack, and finally destroys it.

5 Frequently Asked Questions

I am looking for an implementation of the MRCP server for my ASR/TTS/SVI engine. Should I implement another application based on the libunimrcpserver library?

There is no such a requirement to implement a new application for that. You may use the unimrcpserver application and implement a new plugin for your engine.

When would I need to use the libunimrcpserver library without using the unimrcpserver application?

Suppose you have your own media server application and your goal is to extend it with the MRCP server capabilities. In this case, you may integrate the libunimrcpserver library into your application.

6 References

6.1 Implementation

- unimrcpserver – a server application written in C

6.2 Documentation

- UML Design Concepts – hierarchy, activity and sequence diagrams of the client stack.
- API Reference – a documentation generated by Doxygen from the source code