

T-Engine Forum Specification

TEF050-S401-01.00.00/en
2005-10

UC VoIP Profile Specification

T-Engine Forum VoIP Sub Working Group



Number: TEF050-S401-01.00.00
Title: UC VoIP Profile Specification
Status: Working Draft, Final Draft for Voting, Standard
Date: October, 2005 (First Edition)

Table of Contents

1	Introduction.....	4
1.1	Application.....	4
1.2	Purpose.....	4
2	VoIP Profile Specification.....	4
2.1	Software Configuration.....	4
2.2	VoIP API Specifications.....	4
2.2.1	W ucsp_voip_Open ().....	5
2.2.2	ER ucsp_voip_Close (W vd).....	5
2.2.3	ER ucsp_voip_Connect(W vd, B *addr).....	5
2.2.4	ER ucsp_voip_Disconnect(W vd).....	6
2.2.5	W ucsp_voip_Arrive(W vd).....	6
2.2.6	ER ucsp_voip_Accept(W vd).....	6
2.2.7	ER ucsp_voip_Reject(W vd).....	7
2.2.8	W ucsp_voip_Hold(W vd, W flag).....	7
2.2.9	ER ucsp_voip_State(W vd, UCSP_VOIP_INF *info).....	7
2.3	VoIP Middleware Initial Setting File.....	9
2.3.1	VoIP_MW.ini.....	9
2.3.2	VoIP_Config.ini.....	9
3	Reference Data.....	9

1 Introduction

1.1 Application

This specification specifies the profile specification (API) for VoIP (Voice over IP) software implementation on Ubiquitous Communicator (It is abbreviated as “UC” hereafter).

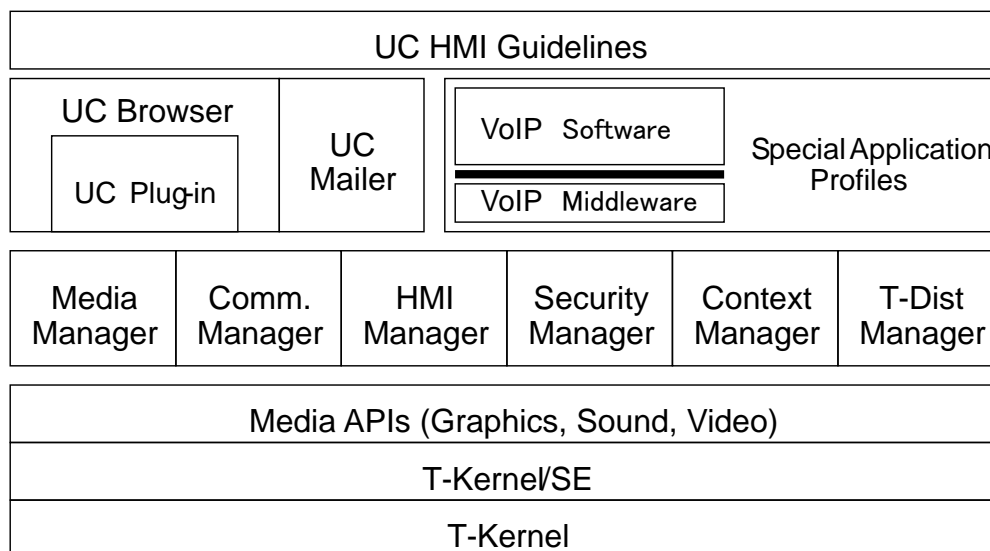
1.2 Purpose

This specification is intended to clarify VoIP Software profile (= API of middleware) on UC and facilitate the implementation/porting.

2 VoIP Profile Specification

2.1 Software Configuration

The software configuration specified by this specification and the API which specifies the specifications are shown as below.



API Specification specified by this specification

This specification describes the specification of API when VoIP software is created by using the VoIP middleware provided by each company.

In addition, the VoIP middleware shall include calling sequence and voice codec.

2.2 VoIP API Specification

This paragraph describes the details of function set which is provided as an application interface by VoIP middleware.

The return value and an error code conform to T-Kernel and an error code of T-Kernel Extension. However, they may be extended according to the implementation configuration of the VoIP middleware.

2.2.1 W ucsp_voip_Open ()

Start a procedure to secure the resource necessary for VoIP.

Argument)

Non-existent

Return value)

≥ 0 : Value of descriptor

< 0 : Error

2.2.2 ER ucsp_voip_Close (W vd)

Start a procedure to release the resource secured for VoIP.

Meanwhile, the secured resource shall not be released unless this API is explicitly called.

Argument)

W vd : Descriptor

Return value)

$= 0$: Completed successfully

< 0 : Error

2.2.3 ER ucsp_voip_Connect (W vd, B *addr)

Start a procedure to establish the call session.

Meanwhile, the address configuration of intended party is middleware implementation-dependent. For example, the address configuration shall be SIP-URI when a calling sequence is executed using SIP.

Argument)

W vd : Descriptor

B *addr : Address of intended party

Return value)

$= 0$: Completed successfully

< 0 : Error

2.2.4 ER ucsp_voip_Disconnect(W vd)

Start a procedure to disconnect the call session.

Argument)

W vd : Descriptor

Return value)

= 0 : Completed successfully

< 0 : Error

2.2.5 W ucsp_voip_Arrive(W vd)

Check if there is a receiving or not.

Meanwhile, checking of a receiving has 3 middleware implementation-dependent configurations: they are the event notification, the interrupt, and the polling. However, this specification does not clearly specify which shall be used among these 3 configurations. The specification of these configurations is to be specified at the configuration file described in 2.3.1.

Meanwhile, all configurations are not always supported, depending on the implementation of middleware.

Argument)

W vd : Descriptor

Return value)

= 0 : Not received

= 1 : Received

< 0 : Error

2.2.6 ER ucsp_voip_Accept(W vd)

Start the receiving process to make a call.

Argument)

W vd : Descriptor

Return value)

= 0 : Completed successfully

< 0 : Error

2.2.7 ER ucsp_voip_Reject(W vd)

Start the receiving process to reject a call.

Argument)

W vd : Descriptor

Return value)

= 0 : Completed successfully

< 0 : Error

2.2.8 W ucsp_voip_Hold(W vd, W flag)

Switch the state of holding and non-holding during a call

Meanwhile, when the line on hold is cut, "1" shall be returned as the return value in the release of holding.

Argument)

W vd : Descriptor

W flag : Holding flag (0: Release of holding 1: Start of holding)

Return value)

= 0 : Completed successfully

= 1 : When the line on hold is cut

< 0 : Error

2.2.9 ER ucsp_voip_State(W vd, UCSP_VOIP_INF *info)

Obtain the information regarding the call state

Argument)

W vd : Descriptor

UCSP_VOIP_INFO *info : Structure which stores the information regarding VoIP state.

Return value)

= 0 : Completed successfully

< 0 : Error

" UCSP_VOIP_INFO" structure which stores the information regarding the state has the following members. Meanwhile, the value to be set to each member differs according to VoIP state.

```
typedef struct{
    W time;           : Talk time (Sec)
    B *addr;         : Address of intended party
    B *state;        : VoIP state
} UCSP_VOIP_INFO ;
```

The followings shall be defined as for the state.

```
UCSP_VOIP_WAITING      : Waiting state
UCSP_VOIP_CONNECTING  : State during the call process
UCSP_VOIP_CALLING     : Waiting state for a response from the intended party
UCSP_VOIP_ARRIVING    : State during receiving
UCSP_VOIP_RINGING     : Waiting state for a response to the intended party
UCSP_VOIP_ESTABLISH   : State during a call
UCSP_VOIP_HOLDING     : State of holding
UCSP_VOIP_DISCONNECTING : State during the disconnection process
```

In the case of the following 2 states, the talk time can be obtained.

```
UCSP_VOIP_TALKING
UCSP_VOIP_HOLDING
```

Besides, the time account for talk time will start when the state transits to “UCSP_VOIP_ESTABLISH” and will complete when the state transits to “UCSP_VOIP_DISCONNECTING” .

” 0” is constantly returned to the talk time in the following states.

```
UCSP_VOIP_WAITING
UCSP_VOIP_CONNECTING
UCSP_VOIP_CALLING
UCSP_VOIP_ARRIVING
UCSP_VOIP_RINGING
UCSP_VOIP_RINGING
UCSP_VOIP_DISCONNECTING
```

In the case of the following states, the address of intended party can be obtained.

```
UCSP_VOIP_CONNECTING
UCSP_VOIP_CALLING
UCSP_VOIP_ARRIVING
UCSP_VOIP_RINGING
UCSP_VOIP_TALKING
UCSP_VOIP_HOLDING
```

Meanwhile, in the case of the following states, “NULL” is returned to the address of the intended party.

```
UCSP_VOIP_WAITING
UCSP_VOIP_DISCONNECTING
```


Besides, the address configuration of the intended party shall be identical with the address which is addressable at “ucsp_voip_Connect()”

2.3 VoIP middleware initial setting file

This paragraph describes the initial setting file which is necessary when VoIP application is executed using VoIP middleware.

2.3.1 VoIP_MW.ini

This is the configuration file which describes the fundamental initial setting value in executing VoIP middleware. Among the following values regarding the configuration of advice of receipt, the one that can be set at VoIP middleware is described.

INTERRUPT : OS interrupt
POLLING : Polling
EVENT : Event receive

2.3.2 VoIP_Config.ini

This is the VoIP middleware implementation-dependent configuration file in making a call. It is VoIP middleware-dependent, thus is not a fundamental configuration file.

3 Reference Data

T-kernel Standard Handbook (Newly-Revised Edition)

Compilation supervised by: Mr. Ken Sakamura

Written and edited by: T-Engine Forum