Network Working Group M. Petit-Huguenin
Internet-Draft A. Huckridge
Expires: January 3, 2001 Netergy Networks, Inc.
July 5, 2000

MGCP User Interface Package
draft-petithuguenin-mgcp-ui-pkg-00

Status of this Memo

Copyright Notice

Abstract

This document proposes a new MGCP[2] package to control the user interface of a feature phone.

Table of Contents

1. Introduction

The use of POTS as final equipment for VoIP deployement is only a transitional phase, and phones with a fully controllable user interface will be soon widely available. Until MEGACO/H.248 implementations[3] becomes available, the master/slave protocol of choice for controlling these devices will be MGCP. Hence this document defines a MGCP package fitted for handling the user interface part of feature phones.

2. User Interface Package

Package Name: U

Symbol	Definition	R	S	Duration
inf	Display Information	x		
pos	Cursor Position	x	BR	
dis	Display Text	x	BR	
clr	Clear Text		BR	
kd	Key Down	x		
ku	Key Up	x		
im	Indicator Mode	x	OO	
rm	Ringer Mode	x	TO	
tn	Tone	x	TO	
oc	Operation Complete	x		

The definition of the user interface package is as follows:

Display Information:

This event is received in an EventState parameter following on an AuditEndpoint command with a RequestedInfo parameter . This event contains the characteristics of the display, i.e. the maximum number of rows and columns.
This example shows the event received for a text display with 2 rows and 20 columns:

```
ES: U/inf(2,20)
```

Cursor Position:

The cursor position is received in an EventState parameter following on an AuditEndpoint command with a RequestedInfo parameter.
This example shows the event received for a cursor positioned on the upper left corner of the screen:

```
ES: U/pos(0,0)
```

The cursor position can also be used as a signal, to modify the current position of the cursor.
This example resets the position of the cursor at the upper left corner of the text display:

```
S: U/pos(0,0)
```

Display Text:

This signal allows notification of the text on the screen. The text is inserted at the current cursor position and is truncated if it wraps to the right side of the display. The cursor position is not modified.
An optional list of styles can be added to the text. These styles apply only for the text supplied with this signal. The example below displays text with a blinking word:

```
S: U/pos(0,0),U/dis("You have a new "),U/pos(0,
15),U/dis(Message,b)
```

The display content can be received in an event following on an AuditEndpoint command, as shown in this example:

```
ES: U/pos(0,0),U/dis("You have a new "),U/pos(0,
15),U/dis(Message,b)
```

Clear Display:

This signal clears entirely the display.

```
S: U/clr
```

Key Down:

This event is sent when a key is pressed.
This example shows the event received when the star key is pressed:

```
O: U/kd(*)
```

Key Up:

This event is sent when a key is released. The duration of the key press is sent with this event.
This example shows the event received when the star key is released 750 milliseconds after having been pressed:

```
O: U/ku(*,750)
```

Indicator Mode:

When received, this event informs the current mode of an indicator. When sent, this signal permits the notification of the

current mode of an indicator. The mode of the indicator can be described either by name (on, off, wink, flash, flutter and brokenflutter) or by a customized cycle description. A customized cycle describes the succession and duration of on/off values for the indicator.
This example shows the activation of the voice-mail indicator in a blinking mode:

```
S: U/im(vm,b)
```

Ringer Mode:
When received, this event informs the current mode of the ringer. When sent, this signal allows the notification of the current mode of the ringer. The ring sequence   can be customized the same way as for the indicator mode.
A timeout value can be specified with the Ringer Mode signal. If so, the ringer will automatically stop and send a OperationComplete event.
This example shows how to play a customized ring:

```
S: U/rm(+(200),-(100),+(50),-(50))
```

Tone:
When received, this event informs the current tone is use. When sent, this signal permits the notification of the current tone. The tone can be described either by name or by a customized description of this tone. Customized tones are described by the succession of frequencies and levels, eventually superimposed and modulated.
A timeout value can be specified with the Tone signal. If so, the tone will automatically stop and send a OperationComplete event. Note that named tones can have an implicit (provisioned) timeout value.

```
S: U/tn
```

Operation Complete:
This event informs that a tone or a ring has timed out. The exact signal that has timed-out is passed as a parameter to this event. Example:

```
O: U/oc(rg(+(512),-(512)))
```

3. Security Considerations

The UI package is subject to the security considerations discussed in the MGCP specification[2].

4. Acknowledgements

Many aspects of the UI package have been retrieved from the H.248 Annex G[4] Document.

References

[1] Crocker, D., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[2] Arango, M., Dugan, A., Elliott, I., Huitema, C. and S. Pickett, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 2705, October 1999.

[3] Cuervo, F., Green, N., Huitema, C., Rayman, A., Rosen, B. and J. Segers, "Megaco Protocol", Internet-Draft, Work In Progress, April 2000.

[4] http://standard.pictel.com/ftp/avc-site/0005_Osa/APC-1852.zip

Authors' Addresses

Marc Petit-Huguenin
Netergy Networks, Inc.
2445 Mission College Blvd.
Santa Clara, CA 95054
US

Phone: +1 408 654 0875
Fax: +1 408 980 0432
EMail: marc.petit-huguenin@netergynet.com

Andy Huckridge
Netergy Networks, Inc.
2445 Mission College Blvd.
Santa Clara, CA 95054
US

Phone: +1 408 654 0946
Fax: +1 408 980 0432
EMail: andy.huckridge@netergynet.com

Appendix A. Formal Syntax Definition

This section provided a formal description of the UI package syntax, following the ABNF syntax defined in RFC2234[1].

```
   Package = PackageEvent / PackageSignal

   PackageEvent = DisplayInformation / CursorPosition / DisplayText /
   KeyDown / KeyUp / IndicatorMode / RingerMode / Tone /
   OperationComplete

   PackageSignal = CursorPosition / DisplayText / ClearText /
   IndicatorMode / RingerMode / Tone

   DisplayInformation = "inf" "(" NumberRows "," NumberColumns ")"

   NumberRows = DIGIT *DIGIT

   NumberColumns = DIGIT *DIGIT

   CursorPosition = "pos" "(" Row "," Col ")"

   Row = DIGIT *DIGIT ; 0 to height - 1

   Col = DIGIT *DIGIT ; 0 to width - 1

   DisplayText = "dis" "(" ( quoted-string / token ) 0*( "," [ Blink /
   Invert / Underline ] ) ")"

   Blink = "b"

   Invert = "i"

   Underline = "u"

   ClearDisplay = "clr"

   KeyDown = "kd" "(" Key ")"

   KeyUp = "ku" "(" Key "," Duration ")"

   Key = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9" /
   "*" / "#" / Redial / Hold / Transfer / ForwardAll / ForwardBusy /
   ForwardNoAnswer / VoiceMail / Line / Function

   Redial = "r"

   Hold = "h"

   Transfer = "t"

   ForwardAll = "fa"

   ForwardBusy = "fb"
```

```
   ForwardNoAnswer = "fna"

   VoiceMail = "vm"

   Line = "l" DIGIT * DIGIT

   Function = "f" DIGIT *DIGIT

   IndicatorMode = "im" "(" Key "," IndicatorModeValue ")"

   IndicatorModeValue = On / Off / Wink / Flash / Blink / CustomMode

   On = "+"

   Off = "-"

   Wink = "w" ; equivalent to +(448),-(64)

   Flash = "f" ; equivalent to +(32),-(32)

   Blink = "b" ;  equivalent to +(512),-(512)

   CustomMode = CustomState 0*{ "," CustomState }

   CustomState = On "(" Duration ")" / Off "(" Duration ")

   Duration = DIGIT *DIGIT ; in milliseconds

   RingerMode = "rm" [ "(" 0*{ CustomMode } [ "," Duration ] ")" ]

   Tone = "tn" "(" ( Dial / InsideDial / StutterDial / SecondDial /
   Ringback / SpecialRingback / Busy / Congestion / SpecialInformation
   / Reorder / RecorderWarning / RecorderDetected / CallerWaiting /
   CallWaiting / CustomToneSequence ) [ "," Duration ] ")"

   Dial = "d"

   InsideDial = "id"

   StutterDial = "sd"

   SecondDial = "2d"

   Ringback = "rb"

   SpecialRingback = "sr"

   Busy = "b"
```

```
   Congestion = "c"

   SpecialInformation = "si"

   Reorder = "r"

   RecorderWarning = "rw"

   RecorderDetected = "rd"

   CallerWaiting = "wc"

   CallWaiting = "cw"

   CustomToneSequence = CustomToneComplex 0*{ "," CustomToneComplex }

   CustomToneComplex = CustomTone / "(" CustomTone 0*{ "," CustomTone }
   ")"

   CustomTone = Frequency "@" Level [ "*" Modulation ]

   Frequency = DIGIT *DIGIT ; in hertz

   Level = "0" / "-" DIGIT *DIGIT ; in DBm0, from 0 to -63 dBm0

   Modulation = DIGIT *DIGIT ; in hertz

   OperationComplete = "oc" "(" RingerMode / Tone ")"
```

Appendix B. TODO

- Checking consistency with H.248 Annex J.
- Adding key labeling.
- Adding softkeys.
- Adding multilanguage/unicode support.
- Considering adding hookswitch key.
- Completing the named tones list.
- Adding support for pixel-addressable display device.

Full Copyright Statement

Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.