

ConfBridge

ConfBridge¹⁾ ist das „Universaltool“ für Konferenzen.

Usage

```
app.ConfBridge(conference,[bridge_profile,[user_profile,[menu]]])
```

Der Konferenzraumname (conference) kann Alpha-Numerisch sein. Die Konferenzraum- und Teilnehmerprofile werden entweder aus der confbridge.conf entnommen²⁾ oder können mit der CONFBRIDGE Dialplanfunktion³⁾ dynamisch gesetzt werden.

```
app.Answer()
app.ConfBridge('k123')
```

Leistung

Kerne	Konferenzen	Teilnehmer pro Konf	Σ Calls
8	100	10	1000

Statische Konferenzräume

In der Defaultinstallation, der UCware, beinhaltet die /opt/ucware/etc/asterisk/confbridge.conf sinnvolle Defaultprofile. Diese sind default_user, default_bridge, default_member und default_room. Weitere Profile können, bei bedarf, hinzugefügt werden.

Statische Konferenzräume müssen als sog. „Custom-Extension“ angelegt werden. Diese werden in den beiden Dateien /etc/ucware/asterisk/extensions-to-internal-custom.lua und /etc/ucware/asterisk/extensions-from-internal-custom.lua definiert. Nach dem Anlegen der beiden Dateien ist es zwingend erforderlich dass der Asterisk neu gestartet wird. Ferner ist darauf zu achten, dass vorher die AEL-Custom-Extensions (/etc/ucware/asterisk/extensions-to-internal-custom.ael und /etc/ucware/asterisk/extensions-from-internal-custom.ael) entfernt wurden.

Wichtig: Die Custom-Extension müssen grundsätzlich lauffähigen Lua-Code enthalten. Ein Fehler, z.B. im Syntax, o.ä., führt dazu, dass der Dialplan bei *jedem* Anrufversuch fehlschlägt!

Weiter Informationen: <https://wiki.asterisk.org/wiki/display/AST/Lua+Dialplan+Configuration>

Beispiel Extensions

/etc/ucware/asterisk/extensions-to-internal-custom.lua:

```
local Channel = require 'core.channel'

local function conference(confpin)
    return function(context, extension)
        Channel["CHANNEL(language)"]="de"
        app.Answer()
        app.Wait(0.5);
        app.Read('pin', 'conf-getpin', 0)
        local pin = Channel.pin
        if pin == confpin then
            app.Confbridge(extension);
        else
            app.Playback('conf-invalidpin')
        end
        app.Hangup();
    end
end

local _M = {}
_M.extensions = {
    ['to-internal-custom'] = {
        ["0815"] = conference('0815');
        ["4711"] = conference('1234');
        -- Konferenzen 123[0-9] mit der Pin „Konferenzraum + 1“
    }
}
```

```

    ["_123X"] = function(context, extension) return conference(tostring(extension) .. '1') end;
--    ["<EXT>"] = conference('<PIN>');
}
}

return _M

```

/etc/ucware/asterisk/extensions-from-internal-custom.lua:

```

local Channel = require 'core.channel'

local _M = {}
_M.extensions = {
    ['from-internal-custom'] = {
        include = {'to-internal-custom'}
    }
}

return _M

```

Komplexes Beispiel mit Datenbank

Das hier aufgeführte Beispiel basiert auf einer Custom-Entwicklung für einen Kunden. In den Konferenzräumen soll es einen Konferenzleiter, master, und normale Teilnehmer, minion, geben. Die Konferenz ist nur „offen“, d.h. die Teilnehmer können miteinander reden, wenn der Konferenzleiter anwesend ist. Die Teilnehmer sollen aber – vor der Anwesenheit des Konferenzleiters – schon beitreten können.

Anlegen der /opt/ucware/etc/asterisk/confbridge.conf:

```

[general]

[default_user]
type=user
dsp_drop_silence=yes

[minion]
type=user
dsp_drop_silence=yes
wait_marked=yes
music_on_hold_when_empty=yes
announce_user_count=yes

[master]
type=user
marked=yes
dsp_drop_silence=yes
music_on_hold_when_empty=yes
announce_user_count=yes

[default_bridge]
type=bridge
record_conference=no

[defconf]
type=bridge

```

In der Asterisk-CLI module load app_confbridge.so ausführen.

Wenn die Datei confbridge.conf existiert, sollte app_confbridge.so beim start von Asterisk automatisch geladen werden. Daher ginge auch ein Asterisk neustart.

Dialplan

Hier eine Beispielhafte Anwendung von ConfBridge, /etc/ucware/asterisk/extensions-from-internal-custom.lua. Die Konferenzräume haben Teilnehmer (minion), welche nur untereinander sprechen sollen, wenn Konferenzleiter (master) anwesend ist.

```

local Channel = require 'core.channel'

```

```

local gsdb = require 'core.gsdb'
local Log = require 'core.logger'

local _M = {}
_M.extensions = {
    ["from-internal-custom"] = {
        ["_93xxx"] = function(context, extension)
            local db = gsdb.getInstance()
            Channel["CHANNEL(language)"]="de"
            local room = string.sub(extension, 3)
            local conf = db.queryFetchOne(
                ('SELECT pin, adminpin from custom_conferences WHERE room = %s;'):format(room)
            )

            app.Answer()
            app.Wait(1)
            if conf then
                for i=1,3 do
                    app.Read('pin', 'conf-getpin', 0)
                    local pin = Channel.pin
                    if pin == conf.adminpin then
                        app.ConfBridge(room, 'defconf', 'master')
                        break
                    elseif pin == conf.pin then
                        app.ConfBridge(room, 'defconf', 'minion')
                        break
                    else
                        Log:verbose("Custom conferences: invalid pin for room " .. tostring(room))
                        app.Playback('conf-invalidpin')
                    end
                end
                app.Playback('vm-goodbye')
            else
                Log:verbose("Custom conferences: no conference found. Requested room " ..
tostring(room))
                app.Playback('conf-invalid')
            end
            app.Hangup();
        end;
    }
}

return _M

```

Datenbank für die Konferenzen.

```

CREATE TABLE IF NOT EXISTS custom_conferences( room VARCHAR(10) UNIQUE, pin VARCHAR(10), adminpin
VARCHAR(10) );

REPLACE INTO custom_conferences (room, pin, adminpin) VALUES
('301', '123', '321'),
...;

```

Weiteres

- <https://wiki.asterisk.org/wiki/display/AST/ConfBridge>

¹⁾
<https://wiki.asterisk.org/wiki/display/AST/ConfBridge>

²⁾
<https://wiki.asterisk.org/wiki/display/AST/ConfBridge+Configuration>

³⁾
<https://wiki.asterisk.org/wiki/display/AST/ConfBridge+Functions>

Last update: 07.06.2023
15:42

archiv:4_x:server:knowledgebase:confbridge https://wiki.ucware.com/archiv/4_x/server/knowledgebase/confbridge

From:

<https://wiki.ucware.com/> - **UCware-Dokumentation**

Permanent link:

https://wiki.ucware.com/archiv/4_x/server/knowledgebase/confbridge

Last update: **07.06.2023 15:42**