



## **Mod Khomp: Manual**

# Sumário

<b><u>Considerações iniciais</u></b> .....	<b>1</b>
<b><u>Configuração</u></b> .....	<b>2</b>
<u>Configuração da API K3L</u> .....	2
<u>Configuração do Endpoint</u> .....	2
<u>Configuração do FreeSWITCH</u> .....	6
<b><u>Comandos de console</u></b> .....	<b>18</b>
<b><u>Recursos adicionais</u></b> .....	<b>20</b>
<u>Aplicações (applications) e canais</u> .....	20
<b><u>Códigos e significados</u></b> .....	<b>26</b>
<u>Estados de canais</u> .....	26
<u>Estados de chamada</u> .....	27
<u>Estados da chamada FreeSWITCH</u> .....	27
<u>Códigos GSM</u> .....	27
<b><u>Solução de problemas</u></b> .....	<b>32</b>
<u>Erro durante a instalação do módulo de kernel</u> .....	32
<u>Configurando parâmetros especiais de sinalização ou áudio</u> .....	33
<u>Inicialização automática dos serviços e módulos de kernel</u> .....	34
<b><u>Apêndice</u></b> .....	<b>35</b>
<u>Disposição dos arquivos</u> .....	35

## Considerações iniciais

Este documento aborda informações sobre o *Endpoint* da Khomp como um todo, desde opções disponíveis de configuração, os *applications* disponibilizados, os possíveis comandos de **CLI**, entre outros.

Para procedimentos sobre a instalação, favor consultar o [README do Endpoint](#).

# Configuração

Configurar o Endpoint da Khomp é uma tarefa que consiste de três etapas:

- Configuração das placas, através da K3L;
- Configuração do Endpoint;
- Configuração do FreeSWITCH.

Estas etapas são descritas mais detalhadamente abaixo.

## Configuração da API K3L

Esta etapa é realizada de maneira semi-automatizada através do programa **khompwizard**, um assistente que configura os parâmetros básicos das placas do sistema. Esse assistente inicializa os arquivos de configuração através de informações obtidas do usuário, quando estas forem necessárias, inicializando as configurações menos utilizadas com os valores padrões.

Normalmente, este programa é executado automaticamente após a instalação do sistema. Entretanto, pode ser necessário executá-lo manualmente caso uma atualização esteja sendo realizada, ou se novas placas foram adicionadas no sistema após a instalação dos *drivers* da placa.

Caso seja necessária a configuração de parâmetros avançados da placa e/ou da sinalização, o programa **k3lconfig** permite acessar todas as configurações disponíveis de cada placa instalada. Para maiores informações sobre este programa, consulte a documentação do mesmo. Para soluções de problemas de sincronismo, consulte a seção [Solução de problemas](#) sobre o procedimento de configuração manual das placas.

## Configuração do Endpoint

A configuração padrão do sistema costuma atender à maior parte das necessidades. Entretanto, as configurações do Endpoint da Khomp podem ser modificadas através do arquivo de configuração `'/usr/local/freeswitch/conf/autoload_configs/khomp.conf.xml'`.

A lista de opções é a seguinte:

### <channels>

Define as configurações gerais de todos os canais da Khomp.

**Sintaxe:** `<param name="..." value="..."/>`

- **dialplan:** Nome do módulo de *dialplan* em uso.
- **auto-fax-adjustment:** Ativa ("yes") ou desativa ("no") o ajuste automático do canal (desabilitar o cancelador de eco e a supressão DTMF) ao detectar tom de FAX;

- **auto-gain-control:** Ativa ("yes") ou desativa ("no") a ativação do controle automático de ganho (AGC) pelo Endpoint;
- **context-digital:** Contexto de entrada para ligações em placas digitais (o padrão é "khomp-DD-LL", onde "DD" será substituído, no momento da ligação, pelo número do dispositivo, "LL" pelo número do link, "CCC" pelo número do canal e "SSSS" pelo número serial do dispositivo);
- **context-fxo:** Contexto de entrada para placas FXO (o padrão é "khomp-DD-CC", onde "DD" será substituído, no momento da ligação, pelo número do dispositivo, "CC" pelo número do canal, e "SSSS" pelo número serial do dispositivo);
- **context-fxs:** Contexto de entrada para placas FXS (o padrão é "khomp-DD-CC", onde "DD" será substituído, no momento da ligação, pelo número do dispositivo, "CC" pelo número do canal, e "SSSS" pelo número serial do dispositivo);
- **context-gsm-call** (ou "context-gsm"): Contexto de entrada para ligações GSM (o padrão é "khomp-DD-CC", onde "DD" será substituído no momento da ligação pelo número do dispositivo, "CC" pelo número do canal, e "SSSS" pelo número serial do dispositivo);
- **context-gsm-sms:** Contexto de entrada para SMSs (o padrão é "khomp-sms-DD-CC", onde "DD" será substituído pelo número de dispositivo, "CC" pelo número do canal, e "SSSS" pelo número serial do dispositivo);
- **context-pr:** Contexto de entrada para ligações em placas KPR (o padrão é "khomp-DD-CC", onde "DD" será substituído, no momento da ligação, pelo número do dispositivo, "CC" pelo número do canal);
- **delay-ringback-co:** Define o tempo de *delay* para ativar a geração de tom de controle de chamada (*ringback*) pelo Endpoint da Khomp quando há uma indicação de *ringback*, e não há áudio sendo enviado por quem indicou a situação de controle da chamada;
- **delay-ringback-pbx:** Define o tempo de *delay* para ativar a geração de controle de chamada (*ringback*) pelo Endpoint da Khomp quando há uma indicação de *ringback*, e o áudio de controle enviado não possui nenhum tom (ou seja, está em silêncio);
- **disconnect-delay:** Define o tempo em milissegundos para realizar o processamento de um evento de desconexão, para ignorar situações onde outros equipamentos realizam o duplo atendimento para derrubar chamadas a cobrar;
- **drop-collect-call:** Ativa ("yes") ou desativa ("no") o derrubamento de chamadas à cobrar. Caso ativo, todas as chamadas à cobrar serão derrubadas não importando o que foi ajustado na variável `KDropCollectCall` (o valor padrão é "no");
- **echo-canceller:** Ativa ("yes") ou desativa ("no") o cancelamento de eco automático pelo Endpoint;
- **flash-to-digits:** Define os dígitos para serem enviados quando o FLASH é detectado na FXS;
- **fxo-send-pre-audio:** Quando ativada ("yes") libera canal de áudio sainte antes da conexão da chamada em placas KFXO (o valor padrão é "yes");
- **fxs-digit-timeout:** Define o *timeout*, em segundos, entre dígitos na FXS;
- **fxs-global-orig:** Número inicial para numeração seqüencial de ramais das placas **KFXS** que não estiverem listadas na seção **<fxs-branches>** (a numeração segue ordem crescente por número da placa e número do canal físico) (o padrão é "0");
- **fxs-co-dialtone:** Seqüências de números, separados por vírgula, que disparam um tom contínuo (de central pública) em ramais FXS (ex: "0,99" faz com que, ao discar "0" ou "99", o usuário receba o tom de linha contínuo) (o padrão é vazio);
- **fxs-bina:** Quando ativada ("yes"), ligações para ramais FXS enviarão os dígitos correspondentes ao telefone de origem em sinalização BINA DTMF (o valor padrão é "no");
- **ignore-letter-dtmfs:** Define se o canal deve ignorar DTMFs incomuns detectados pela placa (A, B, C e D). Entretanto, se você necessita passar esses dígitos pela placa, você deve ajustar esta opção para "no" (o valor padrão é "yes");
- **input-volume:** Define o volume de entrada das ligações, varia de -10 a +10 ;

- **kommutter-activation:** Define se a ativação de dispositivos kommuter encontrados no sistema será feita de forma automática ("auto"), ou de forma manual ("manual") pelo usuário, através do comando "khomp kommuter on/off";
- **kommutter-timeout:** Define o *timeout* (em segundos) com que os kommuters serão inicializados. Se chegarem a este *timeout* sem receberem notificação do channel, os dispositivos irão comutar para o estado "desligado". O valor mínimo é "0" , onde os links permanecerão sempre comutados no estado "ligado", e o valor máximo é "255";
- **log-to-console:** Define mensagens de log que devem ser impressas na console;
- **log-to-disk:** Define mensagens de log que devem ser salvar em disco;
- **out-of-band-dtmfs:** Ativa ("yes") ou desativa ("no") a supressão DTMF e o envio destes out-of-band;
- **output-volume:** Define o volume de saída das ligações, varia de -10 a +10 ;
- **pulse-forwarding:** Ativa ("yes") ou desativa ("no") a detecção de pulsos e a conversão dos mesmos em DTMFs;
- **r2-preconnect-wait:** Define o tempo de espera do envio da sinalização de ringback, no protocolo R2/MFC, para iniciar o envio de áudio de silêncio. Apenas utilizado quando "r2-strict-behaviour" estiver ajustado para "no";
- **r2-strict-behaviour:** Ativa ("yes") ou desativa ("no") o comportamento da sinalização R2/MFC conforme a norma define. O padrão é "no", e pode ser alterado para "yes" caso seja necessário receber/enviar dados precisos da sinalização do protocolo (condição de B, por exemplo);
- **suppression-delay:** Ativa ("yes") ou desativa ("no") o delay necessário para supressão DTMF. Se desativado ("no"), também desativa supressão de DTMFs;
- **trace:** Define opções de depuração. Não deve ser utilizado em produção a não ser que estritamente necessário;
- **user-transfer-digits:** Define uma seqüência de dígitos DTMF para iniciar a transferencia entre o FreeSWITCH® e um outro PABX (utilizando sinalização de usuário, como QSig ou FXO FLASH);

## <groups>

Define os grupos para serem usados na alocação de canal.

Neste caso, as opções são usadas para definir nomes para *strings de alocação de canais*. O formato segue o padrão <param name="nome grupo" value="string alocação"/>, onde as *strings de alocação de canais* são as mesmas utilizadas no application bridge, e *nome do grupo* é um nome arbitrário escolhido pelo usuário.

Por exemplo, para definir o grupo **pstn** como os canais 0 e 5 da placa 0, deveria-se utilizar a linha:

```
<param name="pstn" value="b0c0 + b0c5"/>
```

Este grupo, por sua vez, poderia ser usado no *application bridge* como <action application="bridge" data="Khomp/Gpstn/..."/>.

Pode-se também associar um determinado contexto de entrada a um grupo de canais, bastando especificar um nome de contexto após a *string* de alocação, separado por ':' da mesma.

Por exemplo, para definir o mesmo grupo **pstn** acima como os canais 0 até 20 da placa 0, com contexto de entrada **from-pstn**, poderia-se utilizar a linha:

```
<param name="pstn" value="b0c0-20:from-pstn"/>
```

Este grupo, por sua vez, poderia ser usado no *application bridge* como **<action application="bridge" data="Khomp/Gpstn/...">**, e todas as ligações vindas destes canais seriam tratadas no contexto **from-pstn**.

### <cadences>

Define configurações de cadências para o Endpoint.

Neste caso, as opções são nomes de cadências e um ou dois pares de números, que definem os intervalos de tom e silêncio a ser utilizado nas cadências. Para maiores detalhes, favor consultar o arquivo de configuração.

### <fxs-branches>

Define números de origem para a placa KFXS.

Neste caso, as opções são seqüências de prefixos de ramais e números seriais das placas, que definem os números base dos endereços de origem, e a ordem numérica das placas. O formato das opções é:

```
<param name="prefixo" value="serial1, serial2, ..."/>
```

Por exemplo, para definir que as placas K0374 e K2352 devem ser numeradas seqüencialmente, partido do ramal 200, basta escrever:

```
<param name="200" value="374, 2352"/>
```

Para maiores detalhes, favor consultar o arquivo de configuração.

### <fxs-hotlines>

Define hotlines para a placa KFXS.

Neste caso, as opções são seqüências de ramais e números de destino, definindo os ramais listados nesta seção para serem tratados como "hotlines". Para cada ramal listado, o número de destino especificado será discado quando o ramal for retirado do gancho. Exemplo:

```
<param name="100" value="1234"/>  
<param name="200" value="4321"/>
```

No primeiro exemplo, o ramal de número 100 irá telefonar para número 1234 ao ser retirado do gancho; no segundo, o ramal de número 200 irá telefonar para o número 4321 ao ser retirado do gancho.

## <fxs-options>

Permite definir configurações específicas por ramal FXS.

Neste caso, as configurações são números de ramais (baseado nos definidos na seção <fxs-branches>), e as opções e seus valores. As opções disponíveis são:

- context;
- input-volume;
- output-volume;
- calleridnum;
- calleridname;
- flash-to-digits.

Cada opção é separada uma da outra por um pipe "|" ou uma barra "/" e definidas após dois pontos ":", exemplo:

```
<param name="200" value="input-volume:1|context:master-branch" />
```

Para maiores informações sobre a sintaxe e exemplos, favor consultar o arquivo de configuração.

Para maiores informações, consultar o arquivo de configuração '**khomp.conf.xml**'.

## Configuração do FreeSWITCH

Quando as ligações são recebidas nas placas e dispositivos Khomp, estas são encaminhadas pelo *Endpoint* da Khomp para contextos específicos dentro do plano de discagem do FreeSWITCH®. Estes contextos podem ser alterados através do arquivo de configurações **khomp.conf.xml**, disponível no diretório de configuração do FreeSWITCH (por padrão, `/usr/local/freeswitch/conf/autoload_configs`).

Para maiores detalhes sobre os contextos específicos, consultar a seção de [Configuração do Endpoint](#).

Abaixo, encontram-se informações de como configurar os contextos de entrada de chamadas:

### Contextos de entrada em canais E1

Para placas E1, as ligações de entrada por padrão chegam em um contexto pré-definido conforme a opção **context-digital**:

```
<param name="context-digital" value="khomp-DD-LL"/>
```

Este contexto padrão define que as ligações serão redirecionadas de acordo com o número da placa e número do *link*: **DD** é o número dispositivo (com dois dígitos), e **LL** é o número do *link* (também com dois dígitos).

Entretanto, é possível configurar outros contextos de entrada, com formatos diferenciados. Pode-se utilizar a opção **CCC**, que significa o número do canal na placa (com três dígitos), e **SSSS**, que representa o número serial da placa (com quatro dígitos).

#### Exemplos de configuração no arquivo **khomp.conf.xml**:

```
<!-- número seqüencial da placa e do link (ex: khomp-01-00) -->
<param name="context-digital" value="khomp-DD-LL"/>

<!-- número serial da placa e seqüencial do link (ex: khomp-3049-00) -->
<param name="context-digital" value="khomp-SSSS-LL"/>

<!-- número seqüencial da placa e do canal (ex: khomp-00-001) -->
<param name="context-digital" value="khomp-DD-CCC"/>

<!-- recebe todas as chamadas em um só contexto (khomp-digital) -->
<param name="context-digital" value="khomp-digital"/>
```

#### Abaixo um exemplo de contexto no plano de discagem:

```
<!--
Este contexto presente em seu arquivo de dialplan, irá manipular chamadas
de entrada no link 0 (primeiro link) da placa 0.
-->
<context name="khomp-00-00">
    .
    .
    .
</context>
```

#### Outro exemplo, utilizando o mesmo formato:

```
<!--
Este contexto presente em seu arquivo de dialplan, irá manipular chamadas
de entrada no link 1 (segundo link) da placa 0.
-->
<context name="khomp-00-01">
    .
    .
    .
</context>
```

#### Um exemplo completo, com algumas ações simples:

```
<context name="khomp-00-00">
    <extension name="exemplo_1">
        <condition field="destination_number" expression="^1234$">
            <action application="bridge" data="Khomp/b0L1/2345"/>
        </condition>
    </extension>
    <extension name="exemplo_2">
        <condition field="destination_number" expression="^23(\d{2})$">
            <action application="bridge" data="sofia/${use_profile}/11$1@${sip_from_host}"/>
        </condition>
    </extension>
</context>
```

```
<context name="khomp-00-01">
  <extension name="exemplo_3">
    <condition field="destination_number" expression="^1111$">
      <action application="bridge" data="Khomp/b0L0/2345"/>
    </condition>
  </extension>
</context>
```

Este plano de discagem define o recebimento de chamadas na placa **0** e no link **0**, redirecionando chamadas para o número 1234 feitas para o link **1** da placa **0**, pro ramal/telefone **2345**, e redirecionando qualquer número de quatro dígitos começado com **23** para telefones SIP de quatro dígitos começados com **11** mais os dois últimos dígitos recebidos.

Também há mais uma extensão definida, onde as ligações recebidas no link **1** da placa **0** para o número **1111**, são redirecionadas para o link **0** da placa **0**, para o telefone/ramal **2345**.

## Contextos de entrada em canais FXS/FXO/GSM

Da mesma forma que no contexto de placas E1, as ligações de entrada são encaminhadas pelo *Endpoint* para o FreeSWITCH.

O contexto é pré-definido da seguinte forma, conforme o arquivo **khomp.conf.xml**:

```
<param name="context-gsm" value="khomp-DD-CC"/> ;placas GSM
<param name="context-fxs" value="khomp-DD-CC"/> ; placas FXS
<param name="context-fxo" value="khomp-DD-CC"/> ; placas FXO
```

Nestes casos, **DD** é o número dispositivo (com dois dígitos), e **CC** é o número do canal da placa. Pode-se utilizar também **SSSS**, que representa o número serial da placa.

- **IMPORTANTE:** No caso da placa **KGSM**, as ligações entrantes são **sempre** encaminhadas para o *extension "s"* por padrão, visto que o protocolo GSM não identifica o número de destino nas ligações entrantes, apenas o número de origem - quando não omitido.

## Prioridade de contextos na placa FXS

Em ligações originadas a partir de um ramal FXS, o Endpoint driver procura uma extensão válida (dos dígitos discados) após o dígito # ou após o timeout (configurável pela opção **fxs-digit-timeout**, no arquivo **khomp.conf.xml**). Essa busca é realizada no contexto específico do ramal (definido na seção **<fxs-options>** do arquivo de configuração **khomp.conf.xml**), caso não seja configurado, a busca é realizada no contexto definido na opção **context-fxs**;

## Contextos de mensagens SMS (apenas GSM)

Mensagens SMS são recebidas pelo *Endpoint* da Khomp e encaminhadas para o FreeSWITCH como uma ligação normal, mas sem áudio, que possui algumas variáveis ajustadas com informações recebidas na mensagem - para maiores informações sobre estas variáveis, consulte a documentação das variáveis do *Endpoint*. Este contexto também pode ser alterado, de mesma forma que o contextos acima.

Esta ligação entra no seguinte contexto, conforme o arquivo **khomp.conf.xml**:

```
<param name="context-gsm-sms" value="khomp-sms-DD-CC" />
```

Onde **DD** é o número dispositivo (com dois dígitos), e **CC** é o número do canal (também com dois dígitos). Por exemplo:

```
<context name="khomp-sms-00-01">
  <extension name="sms">
    <condition field="destination_number" expression="^s$">
      <action application="log" data="DEBUG KSmsType=${KSmsType}"/>
      <action application="log" data="DEBUG KSmsBody=${KSmsBody}"/>
    </condition>
  </extension>
</context>
```

## Contexto de entrada em canais Khomp\_PR (KPR)

Para estas placas, as ligações de entrada possuem um contexto pré-definido, conforme exemplo abaixo:

```
<param name="context-pr" value="khomp-DD-CC" />
```

Neste caso, **DD** é o número do dispositivo (com dois dígitos), e **CC** é o número do canal da placa.

## Contextos por grupo de chamada

A seção **groups**, no arquivo de configuração **khomp.conf.xml**, pode ser utilizado para definir contextos específicos para determinados grupos de canais.

Esta seção é detalhada na seção **Configuração do Endpoint**.

## Utilização do *application bridge*

O aplicativo (ou *application*) **bridge** é responsável por gerar chamadas no FreeSWITCH a partir de um *dialplan*. Este aplicativo pode ser utilizado para gerar chamadas a partir de diversos tipos de *Endpoints*, sendo que cada *Endpoint* segue um formato específico para definir tanto as opções quanto os canais de comunicações a serem utilizados.

## Campos relativos ao *Endpoint*

Quando utilizado para canais da **Khomp**, a *string* de **bridge** pode ter dois, três ou quatro campos separados por uma barra (/). Algumas *strings* de exemplo:

```
<action application="bridge" data="Khomp/B2L0/32625644"/>
<action application="bridge" data="Khomp/*B2L0/32625644"/>
<action application="bridge" data="Khomp/S0411/99991234"/>
<action application="bridge" data="Khomp/Gpstn/99991234"/>
<action application="bridge" data="Khomp/*Gpstn/99991234"/>
<action application="bridge" data="Khomp/B2C58/32625644/category=4:orig=4855553232"/>
<action application="bridge" data="Khomp/b0c9"/>
<action application="bridge" data="Khomp/b0c1+b0c14"/>
<action application="bridge" data="Khomp/r304"/>
```

Nos cinco primeiros exemplos, temos três campos sendo especificados; no sexto, quatro campos são utilizados; e por fim, nos três últimos exemplos, apenas dois são utilizados.

Sobre os campos utilizados, segue a descrição:

- **1° campo**, 'Khomp': identifica o tipo do *Endpoint* em questão;
- **2° campo**, 'B2L0', 'S0411', 'Gpstn', etc: representa a **Política de Alocação de Canais**;
- **3° campo**, '32625644' e '99991234': são os números de destino, para onde será efetuada a ligação (ausente em placas **KFXS**);
- **4° campo**, 'category=4:orig=4855553232': opções adicionais não-obrigatórias, detalhadas mais à frente.

OBS: A *string* de **bridge** com somente dois campos é específica à ligações para uma placa **KFXS**, onde o destino está atrelado automaticamente ao canal alocado, ou ao ramal especificado.

## Política de alocação de canais

A política de alocação de canais, no módulo da **Khomp**, pode ser especificado na própria *string* de *bridge* ou através de grupos, no arquivo de configuração **khomp.conf.xml**. Para especificar placas, canais e links a serem alocados existe a seguinte sintaxe disponível (considerando X, Y e Z como números quaisquer):

- **bX** -- busca os canais na placa 'X', de maneira crescente.
- **bXLY** -- busca canais no link 'Y' da placa 'X', fazendo uma procura crescente (com relação ao número dos canais).
- **bXcY** -- utiliza apenas o canal 'Y' da placa 'X'.
- **bXcY-Z** -- busca por canais, iniciando do canal 'Y' e indo até o canal 'Z' (inclusive), da placa 'X', de maneira crescente.
- **BXcY-Z** -- idem ao anterior, de maneira decrescente.
- **sX** -- busca os canais na placa de serial 'X', de maneira crescente.
- **sXLY** -- busca canais no link 'Y' da placa de serial 'X', fazendo uma procura crescente (com relação ao número dos canais).
- **sXcY** -- utiliza apenas o canal 'Y' da placa de serial 'X'.
- **sXcY-Z** -- busca por canais, iniciando do canal 'Y' e indo até o canal 'Z' (inclusive), da placa de serial 'X', de maneira crescente.
- **SXcY-Z** -- idem ao anterior, de maneira decrescente.

Para buscar por ramais de placas **KFXS** de acordo com o número do ramal, pode ser utilizada a seguinte sintaxe (considerando X e Y números de ramais válidos):

- **rX** -- busca ramal 'X'.
- **RX** -- equivalente ao anterior.
- **rX-Y** -- busca de ramal 'X' a 'Y', ordem crescente.
- **RX-Y** -- busca de ramal 'X' a 'Y', ordem decrescente.

É interessante notar que apenas a capitalização da letra 'B', 'S' ou 'R' define a ordem de busca dos canais; se minúscula, crescente, e se maiúscula, decrescente.

Já para a alocação de canais através de grupos, existe a seguinte sintaxe disponível:

- **Ggroupname** -- utiliza a string de *bridge* definida ao grupo "groupname" no arquivo de configuração; já detalhado na seção de configuração do Endpoint.
- **ggroupname** -- equivalente ao anterior.

## Agrupando alocações de canais

Existem casos onde é necessário buscar canais mais de um determinado dispositivo, ou determinado grupo de ramais. Para isto, existe uma extensão disponível na string de alocação, que diz respeito ao uso do símbolo de soma (+) para concatenar várias *strings* de ligação, da seguinte forma:

```
<action application="bridge" data="Khomp/B1L0+B2L0/32332933"/>
<action application="bridge" data="Khomp/*B2+B3+B4/99887766"/>
<action application="bridge" data="Khomp/S0411+B1L0/99887766"/>
<action application="bridge" data="Khomp/Gpstn1+Gpstn2/99991234"/>
<action application="bridge" data="Khomp/*gOperadora1+gOperadora2/98891234"/>
```

Esta extensão está disponível tanto no *application bridge* quanto na especificação de grupos, e pode ser utilizada para agrupar qualquer *string* de alocação válida à outra. O processamento das *strings* de alocação se dá da esquerda para a direita - exceto quanto utilizando a alocação cíclica, onde **todos** os canais especificados são verificados simultaneamente.

## Escolha cíclica e/ou justa

Uma variação da alocação de canais é através de uma escolha cíclica e/ou *justa*, que consiste em escolher o canal que **completou** - até o momento - o menor número de ligações **saintes**. Essa forma é caracterizada por um asterisco (\*) antes da string de alocação de canais (conforme pode ser verificado acima, no segundo e quinto exemplos).

Quando iniciada com um asterisco (\*), as outras formas de alocação subsequentes (crescente, decrescente, etc) são utilizadas para decidir, entre os canais com menor número de ligações saintes, qual será verificado primeiro para realizar a chamada.

- **AVISO: O uso da alocação justa e/ou cíclica é recomendável somente em placas analógicas (KFXO), de ramais (KFXS) e de interface celular (KGSM).** Conexões E1 tendem a alocar os canais de

maneira crescente ou decrescente de um lado (na operadora, por exemplo), e o inverso do outro (no PABX, por exemplo), para evitar problemas de dupla ocupação (o que pode ocorrer na sinalização R2/MFC). A alocação cíclica/justa também consome mais recursos de memória e processamento que a alocação tradicional, o que é agravado ainda mais pelo fato de placas E1 terem um número muito maior de canais (30 por link).

Por estes motivos, alocações justas e/ou cíclicas devem ser utilizadas apenas em sinalizações onde isso pode representar alguma diferença real, como equalizar a tarifação das linhas, o uso das mesmas, ou o número de ligações recebidas por cada ponto de atendimento (PA).

## Opções disponíveis

- **orig**: Define o número de origem a ser utilizado na chamada, **sem alterar a variável** `${origination_caller_id_number}`. Ou seja, a opção **orig** serve **apenas** para repassar um número de origem diferente do `${origination_caller_id_number}`. Caso o FreeSWITCH já tenha ajustado a variável `${origination_caller_id_number}`, o que é o comportamento padrão, o Endpoint utiliza este valor automaticamente como referência do número de origem, sem ser necessário repassar nenhuma opção adicional.

Nas placas **KGSM**, se ajustado para **restricted**, omite o número de origem. Exemplo:

```
<action application="bridge" data="Khomp/b0/99887766/orig=restricted"/>
```

- **category**: Quando ajustado para um valor numérico, define a categoria do número de A. Disponível apenas em sinalização R2/MFC;
- **uii**: Quando ajustado para um número e uma string de texto, separados por cerquilha ("#"), envia uma mensagem "User-to-User" para a outra ponta, antes de realizar a chamada, utilizando o descritor como o número e a mensagem como o texto. Disponível apenas em sinalização RDSI (ISDN);
- **ring\_cadence**: Quando ajustado para um identificador, utiliza cadência definida com este nome na seção "cadences" do arquivo de configuração do Endpoint da Khomp para chamar o canal desejado. Disponível apenas em sinalização FXS;
- **ring**: Quando ajustado para dois números, separados por ponto ("."), define as cadências de chamada de um ramal FXS para estes valores, sendo o primeiro relativo ao tempo de chamando, e o segundo, ao tempo de silêncio;
- **ring\_ext**: Quando ajustado para dois números, separados por ponto ("."), define as cadências de chamada extendidas (a serem executadas depois da cadência principal) de um ramal FXS para estes valores, sendo o primeiro relativo ao tempo de chamando, e o segundo, ao tempo de silêncio;
- **usr\_xfer**: Quando ajustado para uma seqüência de dígitos DTMF, define estes como dígitos a serem utilizados para iniciar uma transferência entre PABXes (utilizando sinalizações de usuário);
- **drop\_on**: Quando ajustado para "message\_box", "human\_answer", "answering\_machine", "carrier\_message", "unknown" ou uma lista destes (separados por mais "+" ou ponto "."), faz com que a chamada seja derrubada ao detectar tons de caixa postal, atendimento humano, secretária eletrônica, mensagens da operadora, ou algum atendimento desconhecido - respectivamente. Disponível em sinalizações digitais (links E1 e placas KGSM). Adicionalmente, a informação de atendimento é reportada para o usuário na variável KCallAnswerInfo;
- **answer\_info**: Opção booleana (não necessita de valor). Quando especificada, reporta informações de atendimento para o usuário através da variável KCallAnswerInfo;

- **pre**: Quando ajustado para uma seqüência de dígitos DTMF, utiliza estes para pre-alocar um canal de saída em um PABX analógico, discando o número de B desejado a seguir. Somente disponível para sinalização analógica (FXO);
- **pre\_answer**: Opção booleana (não necessita de valor). Quando especificada, "atende" o canal antes de a ligação ser completada permitindo, por exemplo, que DTMFs possam ser enviados - útil para utilizar em um DISA;
- **output\_volume**: Define o volume de saída da ligação. Varia de -10 a +10;
- **input\_volume**: Define o volume de entrada da ligação. Varia de -10 a +10;

## Lista das variáveis

Segue lista de variáveis disponíveis no Endpoint:

- **KDropCollectCall**: Quando definida antes do atendimento (ao receber uma chamada), ativa ("yes") ou desativa ("no", padrão) o derrubamento de chamadas à cobrar baseado na sinalização recebida da central pública; através do duplo atendimento; ou através do reconhecimento por áudio de uma chamada à cobrar (pode ser definido de maneira global);
- **KR2SendCondition**: Quando definida antes do envio de ringback pelo FreeSWITCH (ao receber uma chamada), ajusta a condição de B para o valor numérico que foi ajustada. Apenas disponível para sinalização R2;
- **KR2GotCategory**: Ajustada pelo Endpoint ao receber uma chamada entrante, e possui a categoria do número de A. Apenas disponível para sinalização R2;
- **KR2GotCondition**: Ajustada pelo Endpoint, e disponível após o retorno de uma chamada realizada pelo FreeSWITCH. Contém a condição de B recebida ao realizar a chamada. Apenas disponível para sinalização R2;
- **KISDNGotCause**: Ajustada pelo Endpoint, e disponível após o retorno de uma chamada realizada pelo FreeSWITCH. Contém o código de *cause* do ISDN recebido ao realizar a chamada. Apenas disponível para sinalização ISDN;
- **KCallAnswerInfo**: Ajustada pelo Endpoint, e disponível após o retorno de uma chamada realizada pelo FreeSWITCH. Contém as informações de atendimento detectadas ao realizar a chamada e está disponível apenas para sinalizações digitais (E1, GSM);
- **KSmsDelivered**: Ajustada pelo Endpoint ao enviar uma mensagem SMS com o *application* **KSendSMS**, e define se a mensagem foi entregue com sucesso ("yes") ou não ("no");
- **KSmsErrorCode**: Ajustada pelo Endpoint ao enviar uma mensagem SMS com o *application* **KSendSMS**, e define o código de erro ao enviar a mensagem;
- **KSmsErrorName**: Ajustada pelo Endpoint ao enviar uma mensagem SMS com o *application* **KSendSMS**, contém o nome do erro em forma textual, ou "None" caso não tenha ocorrido nenhum erro.
- **KSmsType**: Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, define o tipo da mensagem recebida (pode conter os valores "message", "confirm" ou "broadcast");
- **KSmsFrom**: Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, define o número de origem da mensagem recebida (disponível em tipos "message" e "confirm");
- **KSmsDate**: Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, define a data de envio da mensagem (disponível em tipos "message" e "confirm");
- **KSmsSize**: Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o tamanho (em bytes) da mensagem recebida (disponível em tipos "message" e "broadcast");

- **KSmsMode:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o tipo codificação da mensagem recebida (disponível em tipos "message" e "broadcast");
- **KSmsBody:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o texto da mensagem recebida (disponível em tipos "message" e "broadcast");
- **KSmsDelivery:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém a data de entrega de mensagem enviada anteriormente cuja confirmação foi requisitada (disponível no tipo "confirm");
- **KSmsStatus:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o *status* de envio de mensagem anteriormente cuja confirmação foi requisitada (disponível no tipo "confirm");
- **KSmsSerial:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o número de série da mensagem recebida (disponível no tipo "broadcast");
- **KSmsPage:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o número da página relativo à mensagem recebida (disponível no tipo "broadcast");
- **KSmsPages:** Ajustada pelo Endpoint no contexto de entrada das mensagens SMS, contém o número total de páginas a serem recebidas (disponível no tipo "broadcast");
- **KUserInfoDescriptor:** Define/informa descritor do protocolo utilizado na mensagem User-to-User Information (RDSI).
- **KUserInfoData:** Define/informa os dados na mensagem User-to-User Information (RDSI).
- **KFaxSent:** Ajustada pelo Endpoint ao enviar FAX com o application KSendFax, e define se o FAX foi enviado com sucesso ("yes") ou não ("no");
- **KFaxReceived:** Ajustada pelo Endpoint ao receber FAX com o application KReceiveFax, e define se o FAX foi recebido com sucesso ("yes") ou não ("no");
- **KFaxResult:** Ajustada pelo Endpoint ao enviar ou receber FAX com o application KSendFax ou KReceiveFax, respectivamente, e define o seu resultado.

## Descrição das variáveis

Abaixo, segue uma explanação sobre como utilizar as variáveis do Endpoint da Khomp disponíveis no dialplan, tanto para comunicar quanto para receber informações:

### KDropCollectCall

Quando ativada, faz com que o Endpoint da Khomp derrube chamadas a cobrar através de duplo atendimento (disponível para sinalizações 'R2 Digital' e FXO), através da informação disponível no protocolo RDSI e R2/MFC, ou através da detecção do áudio de chamada a cobrar (disponível para qualquer sinalização digital por link E1, e para sinalização GSM).

Esta variável é útil para derrubar chamadas a cobrar para determinados ramais, e deve ser ajustado **obrigatoriamente** antes de realizar qualquer tipo de atendimento - aplicações como **playback** e **bridge** devem ser executadas sempre após ajustar esta variável, por exemplo.

Para melhor funcionalidade, é recomendado também que nenhum estado de chamada (*ringback*) seja enviado antes desta variável ser ajustada, então aplicações devem ser executadas apenas após o ajuste correto desta variável.

Esta variável pode ser ajustada localmente e globalmente, tanto para **yes** quanto para **no**. O ajuste da variável global para **yes** fará com que todas as chamadas a cobrar sejam derrubadas, a não ser que a chamada específica seja ajustada para **no** - isto permite a criação de um filtro global de chamadas a cobrar, com algumas poucas exceções.

Ativando a variável dentro do contexto **default**:

```
<context name="default">
<extension name="exemplo">
.
.
.
<action application="set" data="KDropCollectCall=yes"/>
.
.
.
</extension>
</context>
```

Ativando a variável no contexto global, lembrando que a mesma deve ser configurada no arquivo **vars.xml**:

```
<X-PRE-PROCESS cmd="set" data="KDropCollectCall=yes"/>
```

## KR2SendCondition

Ao receber uma chamada, pode ser definida antes do envio de ringback pelo FreeSWITCH (ou seja, antes do FreeSWITCH executar as aplicações answer, ou bridge). Quando utilizada em sinalização R2/MFC, esta variável ajusta a condição de B para o valor numérico desejado.

Exemplo:

```
<!-- Condição "NUMBER CHANGED", avisa ao chamador que o número de B mudou. -->
<action application="KR2SendCondition" data="3"/>
```

## KR2GotCategory

Ao receber uma chamada, é ajustada pelo Endpoint com a categoria recebida do número que originou a chamada. É ajustada na sinalização R2/MFC, e pode ser consultada em qualquer local do dialplan.

Exemplo:

```
<action application="log" data="DEBUG KR2GotCategory [${KR2GotCategory}]/>
```

## KR2GotCondition

Variável ajustada pelo Endpoint, e disponível após o retorno de uma chamada realizada pelo FreeSWITCH. Contém a condição de B recebida ao realizar a chamada. Disponível apenas para sinalização R2/MFC.

Exemplo:

```
<action application="log" data="DEBUG KR2GotCondition [${KR2GotCondition}]" />
```

## KUserInfoDescriptor

Variável ajustada pelo Endpoint no contexto de entrada, a partir de informações recebidas pela rede RDSI através da funcionalidade User-to-User Information. Contém o número do descritor do protocolo utilizado pela outra ponta, e normalmente contém valor '0', mas este valor é dependente da aplicação utilizada.

Maiores informações, consultar a especificação ITU-T Q931 (mais precisamente, a tabela 4-26 da especificação).

Exemplo (trabalhando com o número do descritor do protocolo):

```
<action application="log" data="DEBUG KUserInfoDescriptor [${KUserInfoDescriptor}]" />
```

## KUserInfoData

Variável ajustada pelo Endpoint no contexto de entrada, a partir de informações recebidas pela rede RDSI através da funcionalidade User-to-User Information. Contém os dados propriamente ditos, que foram recebidos, em forma de uma 'string' de texto.

Maiores informações sobre este recurso, consultar a especificação ITU-T Q931.

Exemplo (trabalhando com os dados recebidos):

```
<action application="log" data="DEBUG KUserInfoData [${KUserInfoData}]" />
```

É importante salientar que as variáveis são sensíveis à capitalização das letras (case sensitive).

## KCallAnswerInfo

Variável ajustada pelo Endpoint em ligações de saída, representando o tipo de atendimento realizado pela outra ponta. Pode conter os seguintes valores:

- ◇ "MessageBox" (\*): detectada caixa postal de um telefone celular;
- ◇ "CarrierMessage": mensagem de operadora enviada antes do atendimento;
- ◇ "AnsweringMachine" (\*\*): atendimento por secretária eletrônica;
- ◇ "HumanAnswer" (\*\*): atendimento humano;
- ◇ "Unknown": tipo de atendimento desconhecido;
- ◇ "Fax": reportado quando um tom de fax for detectado.

(\*) Este tipo de atendimento é detectado por sinais em determinadas frequências que são enviados antes da chamada entrar em uma caixa postal, e variam conforme a operadora. O algoritmo captura a maior parte das caixas postais, mas pode falhar se não existir um sinal claro, ou se o mesmo não estiver dentro dos padrões mais utilizados;

(\*\*) A diferenciação entre estes dois tipos de atendimento depende de configuração específica utilizando o programa **k3lanswerinfoconfig**, sendo a detecção apenas baseada em heurísticas e **nunca** com precisão de 100%.

# Comandos de console

Lista de comandos disponíveis no console do FreeSWITCH para o Endpoint da Khomp:

- **khomp channels disconnect** : Desconecta um ou vários canais. Este comando envia uma mensagem diretamente para o canal físico da placa em questão, requisitando a desconexão. Use com cautela;
- **khomp channels unblock** : Desbloqueia canais bloqueados para entrada ou para saída. Somente disponível em sinalização digital via link E1;
- **khomp clear links**: Limpa os contadores de erros nos links;
- **khomp clear statistics**: Limpa as estatísticas de ligações dos canais, ou as estatísticas de um canal específico;
- **khomp get** : Obtém as opções diversas do Endpoint da Khomp;
- **khomp kommuter** : Ativa ou desativa os kommuters ligados via USB nesta máquina. Somente acessível quando a configuração "kommuter-activation" estiver setada como "manual";
- **khomp kommuter count**: Obtém a quantidade de kommuters ligados via USB nesta máquina;
- **khomp log console**: Ajusta opções de logs no console.
- **khomp log disk** : : Ajusta opções de log em disco.
  - ◆ **khomp log console** e **khomp log disk** dispõem de opções auxiliares **no**, que inverte a escolha de mensagens, e **just**, que generaliza a escolha. Exemplos:
    - ◇ **khomp log disk just commands events** (Habilita *somente* o registro de comandos e eventos da API em disco).
    - ◇ **khomp log disk no commands** (Desabilita o registro em disco de comandos enviados à API).
    - ◇ **khomp log disk warnings** (Habilita *também* o registro em disco dos avisos do Endpoint).
  - ◆ Mais informações sobre as opções de *log* nos comando "**help khomp log disk**" ou "**help khomp log console**".
- **khomp log rotate**: Rotaciona arquivos de log do Endpoint;
- **khomp log status**: Mostra mensagens de log atualmente sendo escritas em disco e mostradas no console;
- **khomp log trace isdn**: Ativa a depuração da sinalização RDSI (ISDN);
- **khomp log trace k3l** : Ativa a depuração de baixo nível da API K3L;
- **khomp log trace r2** : Ativa a depuração de baixo nível da sinalização R2/MFC;
- **khomp reset links**: Envia um comando de reset para um determinado link E1 de uma determinada placa;
- **khomp revision**: Mostra número da versão e revisão do Endpoint;
- **khomp select sim**: Seleciona o SIM card, disponível nas placas KGSM;
- **khomp send command** : Envia comando da API K3L diretamente para a placa. Apenas para depuração de problemas, pode comprometer a estabilidade do sistema se utilizado de maneira incorreta;
- **khomp send raw command** : Envia um comando diretamente para o DSP da placa. Apenas para depuração de problemas, pode comprometer a estabilidade do sistema se utilizado de maneira incorreta;
- **khomp set** : Ajusta opções diversas do Endpoint da Khomp;
- **khomp show calls** : Mostra estados das chamadas Khomp, podendo listar também por placa ou canal específico;
- **khomp show channels** : Mostra o estado dos canais Khomp, podendo listar também por placa específica;
- **khomp show links**: Mostra estados dos links E1 disponíveis.
- **khomp show statistics** : Mostra as estatísticas de ligações dos canais, ou as estatísticas de um canal específico;

- **khomp sms** : Envia uma mensagem SMS utilizando canais da placa KGSM para um determinado número;
- **khomp summary** :Imprime um sumário das placas do sistema e de suas características;

## Recursos adicionais

Este capítulo trata de recursos adicionais do Endpoint, relacionados às funcionalidades especiais presentes em algumas sinalizações.

### Aplicações (applications) e canais

O Endpoint da Khomp, além de registrar um tipo de canal de comunicação "Khomp", registra também os seguintes itens:

#### Aplicação "KUserTransfer"

Realiza o processo de transferência do canal atual para o ramal *número* utilizando o protocolo de sinalização QSig (Single Step Call Transfer) para placas E1 configuradas com sinalização RDSI (ISDN), ou utiliza comando de FLASH para linhas FXO, LineSide, CAS\_EL7 e EILC.

A sintaxe segue:

```
<action application="KUserTransfer" data="número[,opções]" />
```

Exemplo:

```
<action application="answer" />
<action application="KUserTransfer" data="2345" />
```

Os campos têm o seguinte significado:

- **número**: Número para onde a ligação deve ser transferida.
- **opções**: Define as opções de transferência a utilizar, que são:
  - ◆ **n**: Aguarda até o canal ser desconectado.

#### Aplicação "KSendSMS"

Esta aplicação tem a função de enviar mensagens SMS através das placas KGSM da Khomp, utilizando os módulos e SIM cards presentes na placa para tal. A sintaxe da aplicação é a seguinte:

```
<action application="KSendSMS" data="recurso|destino|mensagem" />
```

Podendo cada campo ser resumido em:

- **recurso:** Segue uma sintaxe idêntica à alocação de canais do application Bridge, e define qual modem utilizar;
- **destino:** Número para onde enviar a mensagem, podendo ser precedido ou sucedido por ! para requisitar uma mensagem de confirmação de envio;
- **mensagem:** Texto (sem aspas) que deve ser enviado para **destino**.

Após o envio da mensagem, as variáveis **KSmsDelivered** e **KSmsErrorCode** conterão o resultado do envio da mensagem. Para maiores informações sobre estas, favor consultar a seção sobre as variáveis utilizadas no Endpoint.

Exemplos de uso desta aplicação seguem abaixo:

- Envia "Mensagem de teste." para telefone "99887766" utilizando o modem "1" (segundo modem) da placa "0":

```
<action application="log" data="DEBUG Enviando SMS..." />
<action application="KSendSMS" data="b0c1|99887766|Mensagem de teste" />
```

- Envia "Mensagem de teste." para telefone "99887766" utilizando o primeiro modem livre da placa "0", e verifica retorno do envio:

```
<action application="log" data="DEBUG Enviando SMS..." />
<action application="KSendSMS" data="b0|99887766|Mensagem de teste" />
<action application="log" data="DEBUG Enviou? ${KSmsDelivered}" />
<action application="log" data="DEBUG Codigo: ${KSmsErrorCode}" />
<action application="log" data="DEBUG Descr.: ${KSmsErrorName}" />
```

- Envia "Mensagem de teste." para telefone "99887766" utilizando o primeiro modem livre da placa "0", ou para o primeiro canal livre da placa "1" (se não houver canal livre na primeira placa):

```
<action application="log" data="DEBUG Enviando SMS..." />
<action application="KSendSMS" data="b0+b1|99887766|Mensagem de teste" />
```

- Envia "Mensagem de teste." para telefone "99887766" utilizando o primeiro modem livre da placa "0", requisitando confirmação:

```
<action application="log" data="DEBUG Enviando SMS..." />
<action application="KSendSMS" data="b0|99887766!|Mensagem de teste" />
```

## Aplicação "KEchoCanceller"

Esta aplicação tem a função de habilitar ou desabilitar o cancelador de eco do canal.

```
<action application="KEchoCanceller" data="ação[, opções]" />
```

Onde:

- **ação**: Pode ser **on** para habilitar o cancelador de eco, e **off** para desabilitar;

Exemplo de uso desta aplicação:

```
<action application="KEchoCanceller" data="off"/>
```

## Aplicação "KAutoGainControl"

Esta aplicação tem a função de habilitar ou desabilitar o controle automático de ganho no canal.

```
<action application="KAutoGainControl" data="ação[,opções]"/>
```

Onde:

- **ação**: Pode ser **on** para habilitar o controle automático de ganho, e **off** para desabilitar;

Exemplo de uso desta aplicação:

```
<action application="KAutoGainControl" data="on"/>
```

## Aplicação "KDTMFSuppression"

Esta aplicação tem a função de habilitar ou desabilitar a supressão de DTMF do canal. A sintaxe da aplicação é a seguinte:

```
<action applicatin="KDTMFSuppression" value="ação[,opções]"/>
```

Onde:

- **ação**: Pode ser **on** para habilitar a supressão DTMF, e **off** para desabilitar;

É importante notar que quando desabilitada a supressão de DTMF, os DTMFs serão passados *inband* e não serão mais reportados ao FreeSWITCH. Dessa forma o FreeSWITCH não reconhecerá os DTMFs, o que pode ocasionar em mau funcionamento de aplicações como por exemplo, URAs.

Exemplo de uso desta aplicação:

```
<action applicatin="KDTMFSuppression" value="off"/>
```

## Aplicação "KSetVolume"

Esta aplicação tem a função de ajustar o volume de entrada e saída de canais da Khomp, sendo a sua sintaxe a seguinte:

```
<action application="KSetVolume" data="<volume>"/>
<action application="KSetVolume" data="<output-volume>|<input-volume>"/>
```

Onde os campos possuem o seguinte significado:

- **volume:** Ajusta o volume de entrada e saída (-10 a +10);
- **output-volume:** Ajusta o volume de saída (-10 a +10, "none" para não alterar);
- **input-volume:** Ajusta o volume de entrada (-10 a +10, "none" para não alterar).

## Aplicação "KAdjustForFax"

Esta aplicação tem a função de ajustar um canal da Khomp para o recebimento de sinal de FAX/modem, otimizando o canal de comunicação para o tráfego de dados. Sintaxe:

```
<action application="KAdjustForFax" data=""/>
```

Esta aplicação não recebe parâmetros. Exemplo de utilização:

```
<action application="KAdjustForFax" data=""/>
```

## Aplicação "KSendFax"

Esta aplicação tem a função de enviar fax utilizando canais digitais ou FXO da Khomp em ligações pré-estabelecidas, sendo a sua sintaxe a seguinte:

```
<action application="KSendFax" data="<arquivo>[:<arquivo2>[:...]][|<faxid>"/>
```

**Esta aplicação necessita de uma licença adquirida à parte para ser utilizada em canais digitais.** Os campos possuem os seguintes significados:

- **arquivos:** Arquivos a serem enviados para o fax devem estar encapsulados no formato TIFF e possuírem resolução de 98, 196 ou 392 dpi;
- **faxid:** Número do fax. Caso não seja especificado, o valor será obtido pela id da ligação, e caso esta, também não seja válida, o número do fax será o configurado como padrão na K3L.

Exemplo de uso desta aplicação:

```
<action application="KSendFax" data="/tmp/fax.tif:/home/root/fax2.tif,1234"/>
```

## Aplicação "KReceiveFax"

Esta aplicação tem a função de receber fax utilizando canais digitais ou FXO da Khomp, sendo a sua sintaxe a seguinte:

```
<action application="KReceiveFax" data="<arquivo>[|<faxid>]"/>
```

**Esta aplicação necessita de uma licença adquirida à parte para ser utilizada em canais digitais.** Os campos possuem os seguintes significados:

- **arquivo:** Nome que será atribuído ao arquivo de fax recebido.
- **faxid:** Número do fax. Caso não seja especificado, o valor será obtido pela id da ligação, e caso esta, também não seja válida, o número do fax será o configurado como padrão na K3L.

Exemplo de uso desta aplicação:

```
<action application="answer" />  
<action application="KReceiveFax" data="/tmp/fax.tif"/>
```

## Canal "Khomp\_SMS"

Este canal de comunicação é utilizado para receber mensagens SMS e criar ligações entrantes no FreeSWITCH para cada mensagem recebida. Este canal não possui qualquer tipo de tratamento ou processamento de áudio, e é chamado com cinco variáveis ajustadas:

- **KSmsFrom**, contendo o número de origem de quem enviou a mensagem;
- **KSmsDate**, que define a data/hora do recebimento da mensagem;
- **KSmsSize**, representando o tamanho da mensagem (em bytes);
- **KSmsMode**, contendo a codificação utilizada para transmitir a mensagem;
- **KSmsBody**, que é a mensagem em si.

O processamento do dialplan do FreeSWITCH pode ser utilizado para guardar esta mensagem em um banco de dados, executar alguma aplicação, entre outros. Entretanto, a única ação aceita por este channel é desligamento (hangup) - ou seja, esta ligação entrante não pode ser considerada uma ligação comum.

## Canal "Khomp\_PR"

Este canal de comunicação é utilizado para receber ligações em placas de gravação passiva (família **KPR** e **KFXO-HI**), criando ligações entrantes no FreeSWITCH para cada chamada recebida. Este canal permite apenas o recebimento de áudio capturado do *link*, não permitindo tanto o envio de mensagens de áudio quanto o de sinalizações de controle.

O processamento do dialplan do FreeSWITCH pode ser utilizado para gravar dados sobre esta ligação em um banco de dados, executar alguma aplicação especial e/ou algum *application* de gravação (como o **record**), entre outros. Entretanto, a única ação aceita por este channel é desligamento (hangup) - ou seja, esta ligação entrante não pode ser considerada uma ligação comum.

# Códigos e significados

Este capítulo apresenta os códigos presentes no Endpoint da Khomp e seus significados, utilizados nos comandos de console:

## Estados de canais

Refletem o estado do canal relativo à placa. No caso de links E1, o estado pode ter uma ou mais das seguintes opções:

- **Free:** o canal está livre;
- **Busy:** o canal não está livre (ou ocupado, ou em falha);
- **Outgoing:** o canal possui uma ligação de saída;
- **Incoming:** o canal possui uma ligação de entrada;
- **Locked:** o canal está bloqueado;
- **Outgoing Lock:** o canal está bloqueado para chamadas saíntes;
- **Local Fail:** o canal possui uma falha local (nesta ponta);
- **Incoming Lock:** o canal está bloqueado para chamadas entrantes;
- **Remote Lock:** há um bloqueio remoto (na outra ponta) neste canal.

No caso de um canal FXS, o estado é definido por um destes valores:

- **On Hook:** o telefone conectado neste canal está no gancho ou desconectado;
- **Off Hook:** o telefone conectado neste canal está fora do gancho;
- **Ringin:** o canal está sendo chamado;
- **Failure:** o canal está em falha devido a problemas de comunicação entre a central e a placa.

No caso de um canal GSM, o estado também é definido por um dos valores a seguir:

- **Idle:** o canal está livre e disponível para realizar chamadas;
- **Call In Progress:** o canal está ocupado em uma ligação;
- **Modem Error:** ocorreu um erro na comunicação com o modem do canal;
- **SIM Card Error:** o SIM card não está presente ou não foi inserido/detectado corretamente;
- **Network Error:** ocorreu um erro ao comunicar-se com a rede;
- **Not Ready:** o modem está sendo inicializado no canal.

E no caso de um canal FXO, os estados serão os seguintes:

- **Disabled:** o canal está desabilitado;
- **Enabled:** o canal está habilitado.

## Estados de chamada

Define o estado de cada canal relativo ao software, que pode ser::

- **Free:** o canal está livre;
- **Incoming:** o canal está recebendo uma chamada;
- **Outgoing:** o canal está realizando uma chamada;
- **Failure:** o canal está em falha.

## Estados da chamada FreeSWITCH

Reflete diretamente o estado de chamada controlado pelo FreeSWITCH, que pode se

- **new:** Canal é recém criado;
- **init:** Canal foi inicializado;
- **routing:** Canal está procurando uma extensão para executar;
- **execute:** Canal está executando seu dialplan;
- **ex\_media:** Canal está trocando media com outro canal;
- **cs\_media:** Canal está consumindo toda media;
- **hangup:** O canal está marcada para hangup e pronto para terminar.

## Códigos GSM

Os seguintes códigos numéricos são reportados:

### Códigos de SMS (*SMS cause*)

1	Unassigned number
8	Operator determined barring
10	Call barred
21	SMS transfer rejected
27	Destination out of service
28	Unidentified subscriber
29	Facility rejected
30	Unknown subscriber
38	Network out of order
41	Temporary failure
42	Congestion
47	Resources unavailable
50	Facility not subscribed
69	Facility not implemented
81	Invalid SMS transfer reference value
95	Invalid message

96	Invalid mandatory information
97	Message type non existent
98	Message not compatible with SMS protection state
99	Information element non existent
111	Protocol error
127	Interworking
128	Telematic interworking not supported
129	SMS type zero not supported
130	Cannot replace SMS
143	Unspecified TPPID error
144	Alphabet not supported
145	Message class not supported
159	Unspecified TPDCS error
160	Command cannot be actioned
161	Command unsupported
175	Unspecified TP command error
176	TPDU not supported
192	SC busy
193	No SC subscription
194	SC system failure
195	Invalid SME address
196	Destination SME barred
197	SM rejected duplicate SM
198	TPVPF not supported
199	TPVP not supported
208	SIM SMS storage full
209	No SMS storage capability in SIM
210	Error in SMS
211	Memory capacity exceeded
213	SIM data download error
255	Unspecified error
300	Phone failure
301	SMS service reserved
302	Operation not allowed
303	Operation not supported
304	Invalid PDU mode parameter
305	Invalid text mode parameter
310	SIM not inserted
311	SIM PIN necessary
312	Phone SIM PIN necessary
313	SIM failure
314	SIM busy
315	SIM wrong
320	Memory failure
321	Invalid memory index
322	Memory full
330	SMSC address unknown
331	No network service
332	Network timeout
500	Unknown error
512	Network busy
513	Invalid destination address
514	Invalid message body length
515	Phone is not in service
516	Invalid preferred memory storage
517	User terminated

## Códigos de chamada (*call cause*)

1	Unallocated number
3	No route to destination
6	Channel unacceptable
8	Operator determined barring
16	Normal call clear
17	User busy
18	No user responding
19	No answer from user
21	Call rejected
22	Number changed
26	Non Selected user clear
27	Destination out of order
28	Invalid number format
29	Facility rejected
30	Response status enquiry
31	Normal, unspecified
34	No circuit channel available
38	Network out of order
41	Temporary failure
42	Switch congestion
43	Access information discarded
44	Requested channel unavailable
47	Resource unavailable
49	QoS unavailable
50	Request facility not subscribed
55	Call barred with UG
57	Bearer capability not authorized
58	Bearer capability not available
63	Service not available
65	Bearer capability not implemented
69	Request facility not implemented
70	Only restricted bearer capability available
79	Service not implemented
81	Invalid call reference value
82	User not member of UG
88	Incompatible destination
91	Invalid transit network selected
95	Invalid message
96	Missing mandatory information element
97	Message type not implemented
98	Message incompatible with state
99	Information element not implemented
100	Invalid information element
101	Message incompatible with state (2)
102	Recovery on timer expiry
111	Protocol error
127	Interworking

## Códigos gerais (*mobile cause*)

0	Phone failure
1	No connection to phone
2	Phone adaptor link reserved
3	Operation not allowed
4	Operation not supported

5	Phone SIM PIN required
6	Phone FSIM PIN required
7	Phone FSIM PUK required
10	SIM not inserted
11	SIM PIN required
12	SIM PUK required
13	SIM failure
14	SIM busy
15	SIM wrong
16	Incorrect password
17	SIM PIN2 required
18	SIM PUK2 required
20	Memory full
21	Invalid index
22	Not found
23	Memory failure
24	Text string too long
25	Invalid character in text string
26	Dial string too long
27	Invalid character in dial string
30	No network service
31	Network timeout
32	Network not allowed
33	Command aborted
34	Number parameter instead of text parameter
35	Text parameter instead of number parameter
36	Numeric parameter out of bounds
37	Text string too short
40	Network PIN required
41	Network PUK required
42	Network subset PIN required
43	Network subset PUK required
44	Network service provider PIN required
45	Network service provider PUK required
46	Corporate PIN required
47	Corporate PUK required
60	SIM Service option not supported
100	Unknown
103	Illegal MS #3
106	Illegal MS #6
107	GPRS service not allowed #7
111	PLMN not allowed #11
112	Location area not allowed #12
113	Roaming not allowed #13
132	Service option not supported #32
133	Registration service option not subscribed #33
134	Service option temporary out of order #34
147	Long context activation
148	Unspecified GPRS error
149	PDP authentication failure
150	Invalid mobile class
151	GPRS disconnection TMR active
256	Too many active calls
257	Call rejected
258	Unanswered call pending
259	Unknown calling error
260	No phone number recognized
261	Call state not idle
262	Call in progress
263	Dial state error

264	Unlock code required
265	Network busy
266	Invalid phone number
267	Number entry already started
268	Cancelled by user
269	Number entry could not be started
280	Data lost
281	Invalid message body length
282	Inactive socket
283	Socket already open

## Solução de problemas

Nesta seção, erros e suas soluções mais comuns são apresentados.

### Erro durante a instalação do módulo de kernel

Durante a instalação do *Endpoint* da Khomp, podem ocorrer as seguintes mensagens:

```
K3L: WARNING: Unable to find a module for [...]
```

ou

```
install: ***** THE KERNEL MODULE HAS NOT BEEN INSTALLED: *****
install:
install: ** Please, untar the file kpdriver*.tar.gz located in: **
install: **           '/usr/src/khomp/'           **
install: **           then check the README.txt           **
install: ** for knowing how to proceed with the installation. **
```

Neste caso, será necessário compilar os *drivers* manualmente para o seu sistema. Prossiga para o item abaixo para maiores informações.

### Compilando os *drivers* e iniciando os serviços

Basta seguir ao diretório `/usr/src/khomp`, descompactar o arquivo "kpdriver\_2.0.0XX.tar.gz", e acompanhar procedimentos descritos no arquivo **README.txt**.

Após realizar a compilação e a instalação do módulo, basta carregá-lo no sistema, configurar as placas, e iniciar o servidor de processos da Khomp.

Para carregar o driver de kernel, é necessário executar o seguinte comando:

```
# /etc/init.d/khompdrv start
```

Para configurar as placas, por sua vez, necessário executar o comando:

```
# khompwizard
```

Isto executará um assistente de configuração, que irá perguntar a sinalização utilizada no sistema, bem como outros parâmetros de utilização das placas.

Caso seja necessário configurar outros parâmetros adicionais, pode-se utilizar o seguinte comando:

```
# k3lconfig
```

Este configurador, por sua vez, mostra todas as opções possíveis de configuração da placa. Os parâmetros que não forem configurados assumem os valores padrão automaticamente, e são compatíveis com a maior parte dos

sistemas. Maiores detalhes sobre este programa podem ser obtidos na seção de número '2'.

- **IMPORTANTE:** Para o FreeSWITCH iniciar, é preciso que a placa da khomp esteja configurada e todos módulos estejam rodando (conforme mostrado acima). **Caso a placa não esteja configurada, o FreeSWITCH não iniciará.**

Se você deseja rodar o sistema sem a placa da Khomp, é preciso configurar o FreeSWITCH para ele não carregar o módulo da Khomp. Para isso, abra o arquivo `"/usr/local/freeswitch/conf/autoload_configs/modules.conf.xml"`, e comente a linha que carrega o módulo:

```
<!-- <load module="mod_khomp" /> -->
```

Quando a placa da Khomp estiver devidamente configurada e os módulos da khomp carregados (explicado acima), lembre-se de descomentar esta linha no arquivo.

Por fim, para carregar o servidor de processos, basta executar o seguinte comando:

```
# kserver start
```

Após realizar estes procedimentos, o Endpoint já estará operacional, e o FreeSWITCH já pode ser carregado.

## Configurando parâmetros especiais de sinalização ou áudio

Para configurar parâmetros especiais de sincronismo e/ou sinalização, é possível utilizar o programa "k3lconfig": basta selecionar a placa desejada, e as opções das placas serão apresentadas, divididas em seções e subseções para facilitar o acesso. Não é necessário efetuar a configuração de todos os parâmetros: os valores padrão são assumidos, caso não sejam configurados.

Para ajustar a sinalização do link, basta - depois de selecionar a placa - entrar na seção "Opções de sinalização", e em seguida, em "Sinalização da linha". Para escolher uma sinalização específica, basta utilizar as teclas de direcionamento (setas) até selecioná-la, pressionar 'espaço', e confirmar a opção pressionando 'Enter' sobre o botão "Confirmar".

Por fim, para salvar as configurações modificadas, basta sair do programa: ele irá mostrar uma janela, com opções para salvar ou não as alterações realizadas.

É importante notar que **não é necessário alterar/ativar as seguintes opções:**

- Cancelamento de eco automático;
- Supressão de DTMFs automática;
- Controle de ganho (AGC) automático.

Estas opções são controladas pelo **Endpoint**, e devem estar **desabilitadas** no 'k3lconfig'.

## Inicialização automática dos serviços e módulos de *kernel*

Se a carga dos módulos de *kernel* ou a inicialização dos serviços da Khomp não for realizada automaticamente na inicialização do sistema, é possível realizar esta instalação manualmente, criando um *link* para os *scripts* **/etc/init.d/khompdrv** e **/etc/init.d/kserver** no diretório de inicialização do sistema.

Na caso da distribuição **Debian**, o *script* de carga dos módulos de *kernel* seria *linkado* dentro do diretório */etc/rcS.d/*, enquanto o *script* de inicialização dos serviços seria *linkado* dentro dos diretórios */etc/rc2.d/*, */etc/rc3.d/*, */etc/rc4.d/*, */etc/rc5.d/*, da seguinte forma:

```
# ln -s /etc/init.d/khompdrv /etc/rcS.d/S19khompdrv
# ln -s /etc/init.d/kserver /etc/rc2.d/S20kserver
# ln -s /etc/init.d/kserver /etc/rc3.d/S20kserver
# ln -s /etc/init.d/kserver /etc/rc4.d/S20kserver
# ln -s /etc/init.d/kserver /etc/rc5.d/S20kserver
```

É interessante verificar as normas da sua distribuição para inicializar os serviços de acordo com o que é esperado pela inicialização da mesma.

# Apêndice

Nesta seção, encontram-se informações úteis sobre o Endpoint e componentes relacionados.

## Disposição dos arquivos

Os diretórios criados/modificados nesta instalação são:

```
/etc/init.d/           -- Scripts de inicialização;  
/etc/khomp/           -- Arquivos de firmware e configurações;  
/usr/local/freeswitch/conf/ -- Configurações do FreeSWITCH e Endpoint;  
/usr/doc/khomp/       -- Documentação das placas, do mod_khomp e dos utilitários;  
/usr/sbin/           -- Utilitários e servidor de processos;  
/usr/lib/            -- Bibliotecas compartilhadas da K3L;  
/usr/local/freeswitch/mod/ -- Módulo 'mod_khomp.so';  
/var/log/khomp2.1/    -- Diretório de logs da K3L e Endpoint
```

O *script /etc/init.d/khompdrv* é responsável por carregar os módulos *kpci9030.ko* e *kpex8311.ko* no *kernel*, que deve ser realizada automaticamente na inicialização do sistema. Em caso de problemas, verifique a seção [Solução de problemas](#).