



Channel Khomp: Manual

Sumário

<u>Considerações iniciais</u>	1
<u>Configuração</u>	2
<u>Configuração da API K3L</u>	2
<u>Configuração do channel</u>	2
<u>Configuração do Asterisk</u>	9
<u>Comandos de console</u>	25
<u>khomp channels disconnect</u>	25
<u>Recursos adicionais</u>	29
<u>Aplicações (applications) e canais</u>	29
<u>Interface de gerenciamento (AMI)</u>	38
<u>Interface gateway (AGI)</u>	41
<u>Multiparty em placas KGSM</u>	42
<u>Pré-requisitos</u>	42
<u>Conceitos básicos</u>	42
<u>Chamadas entrantes</u>	43
<u>Novos channels</u>	44
<u>Novos applications</u>	44
<u>Novos comandos</u>	49
<u>Acréscimos no AMI</u>	49
<u>Exemplos de dialplan</u>	50
<u>Compatibilidade com estilo DAHDI de Dialstring</u>	54
<u>Comportamento compatível com DAHDI</u>	54
<u>Exemplos de Dialstrings</u>	54
<u>Utilização de patches adicionais</u>	56
<u>Códigos e significados</u>	57
<u>Estados de canais</u>	57
<u>Estados de chamada</u>	58
<u>Estados da chamada (Asterisk)</u>	58
<u>Códigos GSM</u>	58
<u>Solução de problemas</u>	63
<u>Erro durante a instalação do módulo de kernel</u>	63
<u>Configurando parâmetros especiais de sinalização ou áudio</u>	64
<u>Inicialização automática dos serviços e módulos de kernel</u>	65
<u>Apêndice</u>	66
<u>Disposição dos arquivos</u>	66
<u>Compatibilidade com módulo Zaptel/DAHDI</u>	66

Considerações iniciais

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

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

Configuração

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

- Configuração das placas, através da K3L;
- Configuração do channel;
- Configuração do Asterisk.

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 padrão.

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 channel

A configuração padrão do sistema costuma atender à maior parte das necessidades. Entretanto, a configuração do channel da Khomp pode ser modificadas através do arquivo de configuração **'etc/asterisk/khomp.conf'**.

A lista de opções é a seguinte:

[compatibility]

- **dial-string-like-dahdi**: Ativa a compatibilidade com o DAHDI. Além disso, essa configuração modifica a visualização de alguns comandos do channel da Khomp. (**Disponível a partir da versão: 3.1**);

[channels]

Define configurações gerais de todos os canais da Khomp:

- **accountcode:** Define o account code padrão para chamadas no channel. Esta opção pode ser qualquer string alfanumérica (opção local);
- **amaflags:** Define a flag padrão do Automated Message Accounting, afetando na categorização das entradas no **CDR** do Asterisk (opção local);
- **audio-packet-length:** Define o tamanho dos pacotes de áudio enviados pelo channel para o Asterisk, em bytes (o valor padrão desta opção é 128) (**Disponível a partir da versão: 3.0**);
- **audio-rx-sync:** Define o sistema de sincronização a ser utilizado no tratamento de áudio recebido pela placa, podendo ser ajustado para as seguintes opções (o valor padrão é "auto"):
 - ◆ **board-sync:** cada pacote de áudio recebido pela placa é convertido em uma notificação de novo pacote de áudio para o SoftPBX, sendo apenas possível usar esta opção quando **audio-packet-length** for definido para o mesmo tamanho do pacote de áudio da placa (128bytes);
 - ◆ **softtimer-kernel:** utiliza o mecanismo de temporização do *kernel* disponível pelas chamadas de sistema *timerfd_** para notificação de áudio, recurso este disponível em sistemas Linux *kernel* 2.6.22 ou superior, e *eglibc* ou *glibc* 2.7 ou superior;
 - ◆ **softtimer-thread:** utiliza uma *thread* para notificação de áudio, com temporização baseada em *sleep*;
 - ◆ **auto:** seleciona automaticamente a melhor opção disponível ("board-sync", com *fallback* para "softtimer-kernel", e novo *fallback* para "softtimer-thread").
- **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 (opção local);
- **auto-gain-control:** Ativa ("yes") ou desativa ("no") a ativação do controle automático de ganho (AGC) pelo channel (opção local);
- **callgroup:** Define os grupos padrão de chamada em todos os canais (opção local);
- **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-fxo-alt:** Contexto de entrada alternativo para placas FXO (o padrão é vazio - quando definido, segue a mesma regra de substituição da opção **context-fxo**);
- **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-fxs-alt:** Contexto de entrada alternativo para placas FXS (o padrão é nulo - quando definido, segue a mesma regra de substituição da opção **context-fxs**);
- **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-call-alt** (ou "context-gsm-alt"): Contexto de entrada alternativo para ligações GSM (o padrão é nulo - quando definido, segue a mesma regra de substituição da opção **context-gsm-call**);
- **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-gsm-wait:** Contexto de entrada utilizado para pré-processamento de ligações GSM entrantes que se encontram em estado de espera - para desativar este recurso, utilize **none** (o padrão é "khomp-wait-DD-CC", onde "DD" será substituído, no momento da ligação, pelo número do dispositivo, "CC" pelo número do canal);

- **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);
- **context**: Contexto de entrada para todos os canais de tecnologia **Khomp** (opção local);
- **delay-ringback-co**: Define o tempo de *delay* para ativar a geração de tom de controle de chamada (*ringback*) pelo channel da Khomp quando há uma indicação de *ringback*, e não há áudio sendo enviado por quem indicou a situação de controle da chamada (opção local);
- **delay-ringback-pbx**: Define o tempo de *delay* para ativar a geração de controle de chamada (*ringback*) pelo channel da Khomp quando há uma indicação de *ringback*, e o áudio de controle enviado não possui nenhum tom (ou seja, está em silêncio) (opção local);
- **disconnect-delay**: Define o tempo em milissegundos para realizar o processamento um evento de desconexão, para ignorar situações onde outros equipamentos realizam o duplo atendimento para derrubar chamadas a cobrar (opção local);
- **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") (opção local);
- **echo-canceller** (antigo "echocanceller"): Ativa ("yes") ou desativa ("no") o cancelamento de eco automático do channel (opção local);
- **enabled**: Define se os canais deverão ser usados pelo Asterisk ("yes") ou não ("no"), sendo uma opção útil para depurações, para ser configurado em seções de canais ou grupos específicos. Não deve ser usada em cenários de produção. (opção local) (**Disponível a partir da versão: 3.0**);
- **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");
- **fxo-fsk-detection**: Define o tipo de detecção de *CallerID* a ser realizado, permitindo a detecção de número de origem em padrão **FSK** ou **DTMF** de diversas normas distintas.
NOTA: A detecção de número de origem em formato **BINA DTMF** é realizada pelos DSPs da placa, e está sempre ativada, não sendo afetada por esta configuração.
Para maiores detalhes sobre os valores disponíveis, consulte exemplos do arquivo de configuração. (opção local) (**Disponível a partir da versão: 3.0**);
- **fxo-fsk-timeout**: Define o tempo máximo para detecção do número de origem pelo mecanismo de *CallerID* (o valor padrão é 2000 milissegundos) (opção local) (**Disponível a partir da versão: 3.0**);
- **fxs-digit-timeout**: Define o *timeout*, em segundos, entre dígitos na FXS (opção local);
- **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-send-bina-dtmf** (**fxs-bina** até a versão **3.0 001**): Quando ativada ("yes"), ligações para ramais FXS enviarão os dígitos correspondentes ao telefone de origem em sinalização BINA DTMF (o valor padrão é "yes") (opção local);
- **fxs-send-fsk**: Quando ativada ("yes"), ligações para ramais FXS enviarão os dígitos correspondentes ao telefone de origem em sinalização FSK (o valor padrão é "yes") (opção local) (**Disponível a partir da versão: 3.0**);
- **fxs-sharp-dial**: Quando ativada ("yes"), discar o dígito cerquilha (#) no fim de um número fará com que as ligações procedam imediatamente em ramais FXS, sem aguardar pelo tempo de *timeout* de dígitos,

também permitindo números começados em cerquilha (por exemplo, #1 ou #8) (o valor padrão é "yes") (opção local) (**Disponível a partir da versão: 3.0**);

- **has-ctbus**: Ativa ("yes") ou desativa ("no") o suporte a CT-bus em placas **CTI** (não disponível em modelos **SPX**) (opção local);
- **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") (opção local);
- **input-volume**: Define o volume de entrada das ligações, varia de -10 a +10 (opção local);
- **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" (**Disponível a partir da versão: 2.4.1**);
- **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" (**Disponível a partir da versão: 2.4.1**);
- **language**: Define idioma para ligações nas placas Khomp (opção local);
- **load-error**: Define o comportamento do channel quando ocorre algum erro para inicialização da K3L, que pode ser tanto "skip" (não carrega o channel mas continua a inicialização do Asterisk) ou "failure" (que aborta a inicialização do Asterisk) - o valor padrão é "failure";
- **log-to-console**: Define mensagens de log que devem ser impressas na console;
- **log-to-disk** (antigo "log"): Define mensagens de log que devem ser salvar em disco;
- **mohclass**: Define classe de música de espera para ligações nas placas Khomp (opção local);
- **native-bridge** (antigo "bridge"): Ativa ("yes") ou desativa ("no") o modo de bridge nativo, onde o áudio é comutado diretamente por dentro da mesma placa - caso a ligação esteja situada em dois canais de uma mesma placa e nenhuma opção do Asterisk limite esta comutação direta (o valor padrão é "yes") (opção local);
- **optimize-audio-path**: Define se o canal deve tentar eliminar o atraso de áudio eliminando **agressivamente** pacotes de áudio que não são tratados prontamente pelo Asterisk. Isso garante o atraso mínimo de áudio para o usuário e evita atrasos associados a clientes SIP mal codificados. Entretanto, dependendo da política de escalonamento do sistema, isso pode resultar em descarte excessivo de pacotes e picote no áudio.
IMPORTANTE: Esta opção não deve ser alterada ingenuamente! (o valor padrão é *no*) (opção local).
- **out-of-band-dtmfs** (antigo "dtmfssuppression"): Ativa ("yes") ou desativa ("no") a supressão DTMF e o envio destes out-of-band (opção local);
- **output-volume** (antigo "volume"): Define o volume de saída das ligações. Varia de -10 a +10 (opção local);
- **pickupgroup**: Define o grupo padrão que pode puxar chamadas que estão sendo recebidas (opção local);
- **pulse-forwarding** (antigo "pulsedetection"): Ativa ("yes") ou desativa ("no") a detecção de pulsos e a conversão dos mesmos em DTMFs (opção local);
- **r2-preconnect-wait** (antigo "r2preconnectwait"): 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" (opção local);
- **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) (opção local);

- **record-prefix**: Define o caminho base para os arquivos de gravação das ligações. Caso não seja especificado, o caminho padrão adotado é `"/var/spool/asterisk/monitor/"` (opção local);
- **recording**: Define se a gravação automática deve ocorrer (opção local);
- **suppression-delay** (antigo "suppressiondelay"): Ativa ("yes") ou desativa ("no") o delay necessário para supressão DTMF. Se desativado ("no"), também desativa supressão de DTMFs (opção local);
- **trace**: Define opções de depuração. Não deve ser utilizado em produção a não ser que estritamente necessário;
- **user-transfer-enable**: Define se o canal está habilitado a realizar uma transferência entre o Asterisk® e um outro PABX (utilizando sinalização de usuário, como QSig ou FXO FLASH) (**Disponível a partir da versão: 3.0**) (opção local);
- **user-transfer-digits**: Define uma sequência de dígitos DTMF para iniciar a transferencia entre o Asterisk® e um outro PABX (utilizando sinalização de usuário) (**Disponível a partir da versão: 2.4.1**);
- **qsig-transfer-facility**: Define a *facility* a ser utilizada pelo protocolo ISDN para transferência entre PABXes (para maiores detalhes, consulte exemplos do arquivo de configuração) (opção local) (**Disponível a partir da versão: 3.0**);

NOTA: Opções marcadas como "opção local" podem ser utilizadas em configurações específicas por placa/link/canal/grupo, para maiores detalhes ver informações sobre a seção **[channels-<string>]**.
Disponível a partir da versão: 3.0

[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 **<nome grupo> = <string alocação>**, onde as *strings de alocação de canais* são as mesmas utilizadas no application Dial, 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:

```
pstn = b0c0 + b0c5
```

Este grupo, por sua vez, poderia ser usado no application Dial como **Dial(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:


```
pstn = b0c0-20:from-pstn
```

Este grupo, por sua vez, poderia ser usado no application Dial como **Dial(Khomp/Gpstn/...)**, e todas as ligações vindas destes canais seriam tratadas no contexto **from-pstn**.

[cadences]

Define configurações de cadências para o channel.

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:

```
prefixo = serial1, serial2, ....
```

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

```
200 = 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:

```
100 = 1234  
200 = 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:

- pickupgroup;
- callgroup;
- context;
- input-volume;
- output-volume;
- language;
- mohclass;
- amaflags;
- accountcode;
- calleridnum;
- calleridname;
- mailbox.

Cada opção é separada uma da outra por um pipe "|" ou uma barra "/" e definidas após dois pontos ":". Para maiores informações sobre a sintaxe e exemplos, favor consultar o arquivo de configuração.

[channels-<string>]

Disponível a partir da versão: 3.0

.

Permite definir configurações específicas por agrupamentos de canais, onde <string> segue o mesmo formato das *strings* utilizadas no *application Dial*.

Exemplos:

```
[channels-Gpstn]
echo-canceller = no
auto-fax-adjustment = no
```

```
[channels-b010+b110]
echo-canceller = no
auto-fax-adjustment = yes
```

As opções disponíveis nestas seções estão descritas na seção [channels], marcadas como "opção local".

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

Configuração do Asterisk

Quando as ligações são recebidas nas placas e dispositivos Khomp, estas são encaminhadas pelo *channel* da Khomp para contextos específicos dentro do plano de discagem do Asterisk®. Estes contextos podem ser alterados através do arquivo de configurações **khomp.conf**, disponível no diretório de configuração do Asterisk (por padrão, "/etc/asterisk").

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

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**:

```
context-digital = 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**:

```
; número seqüencial da placa e do link (ex: khomp-01-00)
context-digital=khomp-DD-LL

; número serial da placa e seqüencial do link (ex: khomp-3049-00)
context-digital=khomp-SSSS-LL

; número seqüencial da placa e do canal (ex: khomp-00-001)
context-digital=khomp-DD-CCC

; recebe todas as chamadas em um só contexto (khomp-digital)
context-digital=khomp-digital
```

Abaixo um exemplo de contexto no plano de discagem:

```
; Este contexto presente no extensions.conf irá manipular chamadas
; de entrada no link 0 (primeiro link) da placa 0.
;
[khomp-00-00]
```

Outro exemplo, utilizando o mesmo formato:

```
; Este contexto presente no extensions.conf irá manipular chamadas  
; de entrada no link 1 (segundo link) da placa 0.  
;  
[khomp-00-01]
```

Um exemplo completo, com algumas ações simples:

```
[khomp-00-00]  
exten => 1234,1,Dial(Khomp/b0L1/2345)  
  
exten => _23XX,1,Dial(SIP/11${EXTEN:2})  
  
[khomp-00-01]  
exten => 1111,1,Dial(Khomp/b0L0/2345)
```

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**.

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 *channel* para o Asterisk. Entretanto, estas sinalizações possuem dois contexto pré-definidos, buscados com uma ordem de preferência.

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

```
context-gsm = khomp-DD-CC ; placas GSM  
context-fxs = khomp-DD-CC ; placas FXS  
context-fxo = khomp-DD-CC ; placas FXO
```

Enquanto o segundo contexto possui o seguinte formato padrão:

```
context-gsm-alt = khomp-DD ; placas GSM  
context-fxs-alt = khomp-DD ; placas FXS  
context-fxo-alt = khomp-DD ; 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 *channel driver* procura uma extensão válida dos dígitos discados em três contextos diferentes. A prioridade dos contextos é a seguinte:

1. O contexto específico do canal, que pode ser definido da seguinte forma (a última definição sobrescreverá as anteriores):
 - ♦ Por ramal, definido na seção **[fxs-options]**;
 - ♦ Por canal, definido na seção **[channels]** ou **[channels-<string>]**.
2. Contexto definido na opção **context-fxs**;
3. Contexto definido na opção **context-fxs-alt**.

Caso durante a discagem, nenhuma extensão válida puder ser encontrada, a extensão **i** é procurada nos contextos em ordem de prioridade, e por fim no contexto **default**. Caso ocorra um *timeout* durante a discagem (configurável pela opção **fxs-digit-timeout**, no arquivo **khomp.conf**) durante a discagem, o mesmo comportamento será aplicado - no entanto, a busca se dará inicialmente pelo número já discado até o *timeout* e, se não encontrado, a busca nos contextos será repetida para a extensão "t".

Contextos de mensagens SMS (apenas GSM)

Mensagens SMS são recebidas pelo channel da Khomp e encaminhadas para o Asterisk 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 channel. Este contexto também pode ser alterado, de mesma forma que o contextos acima.

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

```
context-gsm-sms = 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:

```
[khomp-sms-00-01]
exten => s,1,System(/usr/bin/tratar-SMS.sh ${KSmsFrom} ${KSmsBody})
```

Contexto de entrada em canais Khomp_PR (KPR)

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

```
[khomp-DD-CC]
```

Neste caso, **DD** é o número do dispositivo (com dois dígitos), e **CC** é o número do canal da placa. O nome e o formato deste contexto também pode ser alterado através da opção "context-pr" no arquivo de configuração.

Contextos em transferências

O Asterisk é responsável por iniciar transferências, agrupar os dígitos discados e realizar todos os outros passos da lógica de uma transferência de chamada. Entretanto, cada canal alocado no Asterisk (seja para uma ligação saínte ou entrante) permite especificar somente 1 contexto de transferência. Isto implica que somente um contexto (além do default) pode ser usado durante uma transferência de chamada.

Neste caso, o contexto escolhido pelo channel driver segue a seguinte regra para o canal que realizar a transferência:

- Se for o canal que **originou** a chamada, o contexto escolhido é o mesmo contexto que foi usado para originar a chamada corrente;
- Se for o canal que **recebeu** a chamada, o contexto escolhido é o de **maior prioridade**, onde a chamada foi recebida.

Contextos por grupo de chamada

Caso desejado definir contextos específicos para determinados grupos de canais, isto pode ser realizado utilizando a seção **groups**, no arquivo de configuração **khomp.conf**.

Esta opção é detalhada na seção [Configurações do channel](#).

Utilização do *application* Dial

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

Campos relativos ao *channel*

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

```
Dial (KhomP/B2L0/32625644,30,tT)
Dial (KhomP/*B2L0/32625644)
Dial (KhomP/S0411/99991234)
Dial (KhomP/Gpstn/99991234)
Dial (KhomP/*Gpstn/99991234,,t)
Dial (KhomP/B2C58/32625644/category=4:orig=4855553232,,tT)
Dial (KhomP/b0c9,30)
Dial (KhomP/b0c1+b0c14)
Dial (KhomP/r304)
```

Nos cinco primeiros exemplos, temos três campos sendo especificados; no quarto, quatros campos são utilizados;

e por fim, nos três últimos exemplos, apenas dois são utilizados. É importante notar que os valores separados por vírgula (,) são opções do application **Dial**, não do *channel* da **KhomP**.

Sobre os campos utilizados, segue a descrição:

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

OBS: A *string* de **Dial** 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 channel da **KhomP**, pode ser especificado na própria string de Dial ou através de grupos, no arquivo de configuração **khomp.conf**. 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 Dial definida ao grupo "groupname" no arquivo de configuração; já detalhado na seção de configuração do channel.
- 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:

```
Dial (Khomp/B1L0+B2L0/32332933,30,tT)
Dial (Khomp/*B2+B3+B4/99887766)
Dial (Khomp/S0411+B1L0/99887766)
Dial (Khomp/Gpstn1+Gpstn2/99991234)
Dial (Khomp/*gOperadora1+gOperadora2/98891234)
```

Esta extensão está disponível tanto no *application Dial* quanto na especificação de grupos, e pode ser utilizada para agrupar qualquer *string* de alocação válida à outra. O processamento das *strings* de alocação ocorre 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 subseqüentes (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

- **answer_info**: Opção booleana (não necessita de valor). Quando especificada, reporta informações de atendimento para o usuário através do evento AMI **AnswerInfo** e da variável **KCallAnswerInfo** (para acesso à variável, é necessário aplicar um patch no **Asterisk®** ou ajustar a opção **parent** de acordo - não necessário para Asterisk 1.8 ou superior);
- **category**: Quando ajustado para um valor numérico, define a categoria do número de A. Disponível apenas em sinalização R2/MFC;
- **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 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**, sendo necessário aplicar um patch no **Asterisk** para que esta variável possa ser acessada, ou que a opção **parent** seja ajustada para o nome do canal que originou a chamada (não necessário para Asterisk 1.8 ou superior);
- **input_volume**: Define o volume de entrada da ligação, podendo variar de -10 a +10;
- **orig**: Define o número de origem a ser utilizado na chamada, **sem alterar a variável** **_\${CALLERID(num)}**. Ou seja, a opção **orig** serve **apenas** para repassar um número de origem diferente do **_\${CALLERID(num)}**. Caso o Asterisk já tenha ajustado a variável **_\${CALLERID(num)}**, o que é o comportamento padrão, o channel 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:

```
Dial (Khomp/b0/99887766/orig=restricted)
```

AVISO: Em canais de placas KGSM, o único valor válido para esta opção é **restricted**.

- **output_volume**: Define o volume de saída da ligação, podendo variar de -10 a +10;
- **parent**: Ajusta o nome do canal que originou a requisição desta chamada, para permitir que variáveis de retorno sejam ajustadas neste (não necessário para Asterisk 1.8 ou superior);
- **pre**: Quando ajustado para uma sequência 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, "antende" o canal antes de a ligação ser completada permitindo, por exemplo, que DTMFs possam ser enviados - útil para utilizar em um **DISA**;
- **ring_cadence**: Quando ajustado para um identificador, utiliza cadência definida com este nome na seção "cadences" do arquivo de configuração do channel 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 sequência de dígitos DTMF, define estes como dígitos a serem utilizados para iniciar uma transferência entre PABXes (utilizando sinalizações de usuário);
- **uui**: Quando ajustado para um número e uma string de texto, separados por cerquilha ("#"), envia uma mensagem "UserToUser" 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);

Lista das variáveis

Segue lista de variáveis disponíveis no channel:

Nome	Tipo	Descrição
KDropCollectCall	W/O	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, podendo ser definida de maneira global. (Disponível a partir da versão: 2.2)
KR2GotCategory	R/O	Ajustada pelo channel ao receber uma chamada entrante, e possui o código da categoria do número de A. Apenas disponível para sinalização R2.
KR2StrCategory	R/O	Ajustada pelo channel ao receber uma chamada entrante, e possui o nome da categoria do número de A. Apenas disponível para sinalização R2.
KR2GotCondition*	R/O	Ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém o código da condição de B recebida ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de um patch no Asterisk para funcionar corretamente ou que a opção parent seja ajustada corretamente no canal de saída - disponível apenas para sinalização R2.
KR2StrCondition*	R/O	Ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém o nome da condição de B recebida ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de um patch no Asterisk para funcionar corretamente ou que a opção parent seja ajustada corretamente no canal de saída - disponível apenas para sinalização R2.
KISDNGotCause*	R/O	Ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém o código de causa do ISDN, recebido ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de um patch no Asterisk para funcionar corretamente ou que a opção parent seja ajustada corretamente no canal de saída - disponível apenas para sinalização ISDN.
KISDNStrCause*	R/O	Ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém o nome da causa do ISDN recebido ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de

		um patch no Asterisk para funcionar corretamente ou que a opção parent seja ajustada corretamente no canal de saída - disponível apenas para sinalização ISDN.
KCallAnswerInfo*	R/O	Ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém as informações de atendimento detectadas ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de um patch no Asterisk para funcionar corretamente, ou que a opção parent seja ajustada corretamente no canal de saída - disponível apenas para sinalizações digitais (E1, GSM).
KR2SendCondition (deprecated)	W/O	Quando definida antes do envio de ringback pelo Asterisk (ao receber uma chamada), ajusta a condição de B para o valor numérico que foi recebido. Apenas disponível para sinalização R2.
KISDNSendCause (deprecated)	W/O	Quando definida antes do envio de ringback pelo Asterisk (ao receber uma chamada), ajusta a causa para o valor numérico que foi recebido. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KR2Condition	W/O	Quando definida antes do envio de ringback pelo Asterisk (ao receber uma chamada), define a condição de B para o valor numérico que foi ajustada. Apenas disponível para sinalização R2. Disponível a partir da versão: 3.0
KISDNCause	W/O	Quando definida antes do envio de ringback pelo Asterisk (ao receber uma chamada), ajusta a causa para o valor numérico que foi recebido. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KR2Category	R/W	Ajustada automaticamente em chamadas entrantes, contendo a categoria do originante. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta a categoria de A para o mesmo. A variável é automaticamente repassada entre dois canais de sinalização R2, pois é definida de maneira herdável. Apenas disponível para sinalização R2. Disponível a partir da versão: 3.0
KISDNOrigTypeOfNumber	R/W	Ajustada automaticamente em chamadas entrantes, contendo o tipo do número do originador. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o tipo do número do originador para o valor recebido. A variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KISDNDestTypeOfNumber	R/W	Ajustada automaticamente em chamadas entrantes, contendo o tipo do número do destinatário. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o tipo do número do destinatário para o valor recebido. A variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KISDNOrigNumberingPlan	R/W	Ajustada automaticamente em chamadas entrantes, contendo o plano de numeração do originador. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o plano de numeração do originador para o valor recebido. A variável é automaticamente

		repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KISDNDestNumberingPlan	R/W	Ajustada automaticamente em chamadas entrantes, contendo o plano de numeração do destinatário. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o plano de numeração do destinatário para o valor recebido. A variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KISDNOrigPresentation	R/W	Ajustada automaticamente em chamadas entrantes, contendo o tipo de apresentação do número originador. Quando definida com um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o tipo de apresentação do número originador para o valor recebido. A variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN. Disponível a partir da versão: 3.0
KSmsDelivered	R/O	Ajustada pelo channel ao enviar uma mensagem SMS com o <i>application KSendSMS</i> , e define se a mensagem foi entregue com sucesso ("yes") ou não ("no").
KSmsErrorCode	R/O	Ajustada pelo channel ao enviar uma mensagem SMS com o <i>application KSendSMS</i> , e define o código de erro ao enviar a mensagem. <u>Até o channel 2.4.1, contém uma string descrevendo o nome do erro; a partir do channel 3.0, contém o código numérico do erro.</u>
KSmsErrorName	R/O	Ajustada pelo channel ao enviar uma mensagem SMS com o <i>application KSendSMS</i> , contém o nome do erro em forma textual, ou "None" caso não tenha ocorrido nenhum erro (Disponível a partir da versão: 3.0).
KSmsType	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, define o tipo da mensagem recebida (pode conter os valores "message", "confirm" ou "broadcast") (Disponível a partir da versão: 3.0).
KSmsFrom	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, define o número de origem da mensagem recebida (disponível em tipos "message" e "confirm").
KSmsDate	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, define a data de envio da mensagem (disponível em tipos "message" e "confirm").
KSmsSize	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o tamanho (em bytes) da mensagem recebida (disponível em tipos "message" e "broadcast").
KSmsMode	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o tipo codificação da mensagem recebida (disponível em tipos "message" e "broadcast").
KSmsBody	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o texto da mensagem enviada (disponível em tipos "message" e

		"broadcast").
KSmsDelivery	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém a data de entrega de mensagem enviada anteriormente cuja confirmação foi requisitada (disponível no tipo "confirm") (Disponível a partir da versão: 3.0).
KSmsStatus	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o <i>status</i> de envio de mensagem anteriormente cuja confirmação foi requisitada (disponível no tipo "confirm") (Disponível a partir da versão: 3.0).
KSmsSerial	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o número de série da mensagem recebida (disponível no tipo "broadcast") (Disponível a partir da versão: 3.0).
KSmsPage	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o número da página relativo à mensagem recebida (disponível no tipo "broadcast") (Disponível a partir da versão: 3.0).
KSmsPages	R/O	Ajustada pelo channel no contexto de entrada das mensagens SMS, contém o número total de páginas a serem recebidas (disponível no tipo "broadcast") (Disponível a partir da versão: 3.0).
KTDDStringReceived	R/O	Ajustada pelo channel no contexto de entrada das mensagens TDD, contém o texto recebido pela aplicação KReceiveTDD (Disponível a partir da versão: 3.1).
KUserInfoExtended	R/W	Ajustada pelo channel ao receber uma mensagem User-to-User Information (ISDN), define se deverá ser usado o comando estendido (até 256 caracteres) ou padrão (até 32 caracteres) para envio de mensagens UUI . Quando definida para um valor <i>booleano</i> (true ou false), antes da realização de uma chamada pelo Asterisk, define o comando a ser utilizado para envio da mensagem UUI - a variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN.
KUserInfoDescriptor	R/W	Ajustada pelo channel ao receber uma mensagem User-to-User Information (ISDN), contém o descritor do protocolo utilizado na mensagem. Quando definida para um valor numérico, antes da realização de uma chamada pelo Asterisk, ajusta o descritor da mensagem UUI para o valor recebido - a variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN.
KUserInfoData	R/W	Ajustada pelo channel ao receber uma mensagem User-to-User Information (ISDN), contém os dados da mensagem em formato Base 16 (caracteres representados em hexadecimal). Quando definida para uma mensagem em formato Base 16, antes da realização de uma chamada pelo

Asterisk, ajusta os dados da mensagem **UUI** para o valor recebido - a variável é automaticamente repassada entre dois canais de sinalização ISDN, pois é definida de maneira herdável. Apenas disponível para sinalização ISDN.

Retorna o estado relativo ao comando de transferência **KUserTransfer** em sinalização **FXO**, quando a opção **W** é utilizada. Pode estar ajustada para os seguintes valores:

KUserTransferStatus	R/O	<ul style="list-style-type: none">• SUCCESS: Foi detectado atendimento na outra ponta.• TIMEOUT: Não houve atendimento no tempo especificado.• HANGUP: A ligação foi desligada enquanto o atendimento era esperado.
KFaxSent	R/O	Ajustada pelo channel ao enviar FAX com o application KSendFax , e define se o FAX foi enviado com sucesso ("yes") ou não ("no").
KFaxReceived	R/O	Ajustada pelo channel ao receber FAX com o application KReceiveFax , e define se o FAX foi recebido com sucesso ("yes") ou não ("no").
KFaxResult	R/O	Ajustada pelo channel ao enviar ou receber FAX com o application KSendFax ou KReceiveFax , respectivamente, e define o seu resultado.
KOutgoingChannel	R/O	Ajustada pelo channel ao alocar um canal para uma ligação de saída, informando qual placa e canal serão utilizados para efetuar esta ligação. (Disponível a partir da versão: 3.0) .)

Descrição das variáveis

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

KDropCollectCall

Quando ativada, faz com que o channel da Khomp derrube chamadas a cobrar através de duplo atendimento (disponível para sinalização '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 **Dial** 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 como **Progress** ou **Ringin** também 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**:

```
[default]
...
exten => _X.,1,Set(KDropCollectCall=yes)
...
```

Ativando a variável no contexto global:

```
[globals]
...
KDropCollectCall=yes
...
```

KR2Condition

Ao receber uma chamada, pode ser definida antes do envio de ringback pelo Asterisk (ou seja, antes do Asterisk executar as aplicações Answer, Ringing, ou Dial). 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.
;;
exten => _X.,1,Set(KR2Condition=3)
```

KR2GotCategory

Ao receber uma chamada, é ajustada pelo channel 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:

```
exten => _X.,1,NoOp(${KR2GotCategory})
```

KR2GotCondition

Variável ajustada pelo channel, e disponível após o retorno de uma chamada realizada pelo Asterisk. Contém a condição de B recebida ao realizar a chamada. Para versões anteriores ao Asterisk 1.8, necessita de um patch no Asterisk para funcionar corretamente ou que a opção **parent** seja ajustada durante a realização da chamada sainte. Disponível apenas para sinalização R2/MFC.

Exemplo:

```
exten => _X.,1,NoOp( ${KR2GotCondition} )
```

KUserInfoDescriptor

Variável ajustada pelo channel 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):

```
exten => _X.,1,Verbose( ${KUserInfoDescriptor} )
```

KUserInfoData

Variável ajustada pelo channel 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):

```
exten => _X.,1,Verbose( ${KUserInfoData} )
```

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

KCallAnswerInfo

Variável ajustada pelo channel em ligações de saída, quando solicitado pela configuração, 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%.

KOutgoingChannel

Variável ajustada pelo channel em ligações de saída, informando qual canal foi alocado para fazer a ligação, no formato BXCXY, onde X representa o número da placa, e Y o número do canal utilizado.

Obs: Esta variável somente pode ser reportada ao canal originador se o **Dial** informar o parâmetro **parent**, conforme exemplo abaixo:

```
exten => 100,1,Dial(khomp/b010/${EXTEN}/parent=${CHANNEL},30,ht)
```

```
exten => h,1,NoOp(Canal utilizado para fazer a ligação: ${KOutgoingChannel} )
```

KTDDStringReceived

Variável ajustada com a mensagem recebida pela aplicação KReceiveTDD. (**Disponível a partir da versão: 3.1**)

```
exten => 100,1,KReceiveTDD()  
exten => 100,2,NoOp(${KTDDStringReceived})
```

Função CHANNEL()

A função CHANNEL() permite tanto obter quanto ajustar valores no canal em questão. O channel da Khomp suporta os seguintes valores (**Disponível a partir da versão: 3.0**):

Nome	Tipo	Descrição
collectcall	R/O	Retorna "yes" se esta é uma chamada entrante a cobrar, ou "no" se não for (ou se a sinalização não prover esta informação).
gsmantenna	R/O	Retorna o nível de sinal da placa, de 0% a 100%, ou "unknown" se desconhecido.
gsmerrorrate	R/O	Retorna o índice de erros de transmissão de sinal, de 0% a 7%, ou "unknown" se desconhecido.

gsmoperatorname	R/O	Retorna o nome da operadora onde o SIM card está atualmente registrado.
gsmregistrystatus	R/O	Retorna estado de registro do SIM card, que pode ser "notregistered", "registered", "searching", "denied", "roaming", "initializing" ou "unknown".
gsmcard	R/W	Retorna o número do SIM card atualmente em uso, ou ajusta o SIM card para o número inteiro passado como valor.
inputvolume	R/W	Lê ou ajusta o volume de entrada do canal (de -10 a +10).
outputvolume	R/W	Lê ou ajusta o volume de entrada do canal (de -10 a +10).

Exemplo: Lendo valores

```
exten => _X.,1,Set(var_collectcall=${CHANNEL(collectcall)})
```

Exemplo: Configurando valores (variáveis Read/Write ou R/W)

```
exten => _X.,1,Set(CHANNEL(inputvolume)=+5)
```

Comandos de console

Lista de comandos disponíveis no console do Asterisk para o channel da Khomp:

khomp channels disconnect

Desconecta um ou vários canais atualmente conectados. Este comando envia uma mensagem diretamente para o canal físico da placa em questão, requisitando a desconexão. Use com cautela.

khomp calls show

Mostra estados das chamadas Khomp, podendo listar também por placa ou canal específico, em uma tabela contendo informações das ligações (**Disponível a partir da versão: 3.0**).

O identificador (**hw id**) é composto pelo número do dispositivo, número do canal físico (na placa), número do canal lógico e número da chamada. Canais GSM possuem 5 canais lógicos e um canal de conferência, enquanto as outras sinalizações possuem apenas um canal lógico. Todas as sinalizações possuem espaço para duas chamadas por canal lógico.

Os campos **chan** e **call** referem-se ao estado do canal lógico e da chamada, enquanto **orig_addr** e **dest_addr** referem-se ao número de origem e ao número de destino da chamada, respectivamente.

O campo **info** mostra indicações extras sobre o canal, podendo conter:

- **M**: a chamada GSM está em uma conferência **Multiparty**;
- **O**: a chamada possui um canal Asterisk associado.

khomp channels show

Mostra o estado dos canais Khomp, podendo listar também por placa específica.

khomp channels statistics

Mostra as estatísticas de ligações dos canais, ou as estatísticas de um canal específico;

khomp channels unblock

Desbloqueia canais bloqueados para entrada ou para saída. Somente disponível em sinalização digital via link E1.

khomp dump config

Mostra configurações disponíveis (gerais e locais), assim como valores configurados em cada canal na tela
(Disponível a partir da versão: 3.0).

Pode receber um número de dispositivo, ou um número de dispositivo seguido de número de canal, limitando a listagem ao dispositivo ou ao canal especificado, respectivamente.

khomp kommuter

Ativa ou desativa os kommuters ligados via USB nesta máquina. Somente acessível quando a configuração "kommuter-activation" estiver setada como "manual". Disponível a partir da versão: 2.4.1.

khomp kommuter count

Mostra o número de kommuters conectados nesta máquina. Disponível a partir da versão: 2.4.1.

khomp links errors

Mostra estado dos contadores de erro de todos os canais, ou de uma placa/link somente.

khomp links reset

Envia um comando de reset para um determinado link E1 de uma determinada placa.

khomp links show

Mostra estados dos links E1 disponíveis.

khomp log console

Ajusta opções de logs no console.

khomp log disk

Ajusta opções de log em disco.

Dispõem de opções auxiliares **no** - que inverte a escolha de mensagens - e **just**, que faz com que só as mensagens especificadas na linha de comando continuem habilitadas.

Exemplos:

```
khomp log disk just commands events  
; Habilita *somente* o registro de comandos e eventos da API em disco,  
  desativando outras mensagens que estivessem ativadas anteriormente.
```

```
khomp log disk no commands  
; Desabilita o registro em disco de comandos enviados à API.
```

```
khomp log disk warnings  
; Habilita o registro em disco dos avisos do channel.
```

khomp log rotate

Rotaciona arquivos de log do channel.

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 record

Ativa gravação de áudio através do channel.

khomp revision

Mostra número da versão e revisão do channel.

khomp select sim

Seleciona o SIM card disponível nas placas KGSM da Khomp.

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 channel da Khomp, explicadas em maior detalhe na descrição de uso do comando ("help khomp set").

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.

Para maiores informações e exemplos de uso, adicionar **help** antes dos comandos. Por exemplo: **help khomp links errors**.

Recursos adicionais

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

Aplicações (applications) e canais

O channel 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.

A sintaxe segue:

```
KUserTransfer(número[,opções])
```

Exemplo:

```
exten => 1234,1,KUserTransfer(2345)
```

Os campos têm o seguinte significado:

- **número**: Número para onde a ligação deve ser transferida.
 - ♦ Em placas **FXO**, este parâmetro pode ser "**none**" ou omitido, sendo neste caso enviado apenas o comando de transferência (FLASH) para a linha. **Disponível a partir da versão: 3.0**
- **opções**: Define as opções de transferência a utilizar, que são:
 - ♦ **k**: Continua execução do *dialplan* após envio de comando de transferência (apenas disponível em sinalização FXO). **Disponível a partir da versão: 3.0**
 - ♦ **n**: **REMOVIDA NO CHANNEL 3.0, NÃO UTILIZAR**. Aguarda até o canal ser desconectado (apenas disponível em sinalização ISDN).
 - ♦ **t**: Especifica a **facility** do protocolo a ser utilizada para a transferência, podendo conter os mesmos valores que a opção de configuração **qsig-transfer-facility** (apenas disponível em sinalização ISDN). **Disponível a partir da versão: 3.0**
 - ♦ **W**: Aguarda até o canal ser desconectado (apenas disponível em sinalização ISDN).
 - ♦ **W(n)**: Aguarda atendimento ou desconexão por até **n** segundos (apenas disponível em sinalização FXO).
IMPORTANTE: para canais **FXO**, a espera utiliza detecção de voz para validar o atendimento, o que em alguns casos pode exigir ajustes nos parâmetros de detecção de voz da placa. O resultado da execução da aplicação pode ser consultado a partir da variável **KUserTransferStatus**.
Disponível a partir da versão: 3.0

Aplicação "KRecord"

Aplicação para gravar áudio através das placas da Khomp. Esta aplicação foi criada para utilizar o recurso de gravação presente nas placas da Khomp, que pode realizar a mistura do áudio em hardware e enviá-lo diretamente para o disco. A gravação é feita **apenas** em formato **A-Law**, com cabeçalhos **WAV**. Caso o canal de entrada não seja um canal Khomp e a opção **f** não tenha sido informada, o KRecord invoca o *application* **MixMonitor** para realizar a gravação.

A sintaxe da aplicação é a seguinte:

```
KRecord([arquivo[|opções[|outras-opções]])
```

Os campos têm o seguinte significado:

- **arquivo**: Nome do arquivo a gravar; caso omitido, será utilizado um formato de nome padrão.
- **opções**: Define as opções de gravação a utilizar, que são:
 - ♦ **b**: Grava apenas quando canal for conectado (análoga à opção **"b"** do **MixMonitor**);
 - ♦ **f**: Força aplicação **KRecord** a executar, mesmo se o canal de entrada não for do tipo **Khomp**. Neste caso, a aplicação ajusta variáveis do channel atual, para que quando o canal de saída tenha áudio disponível, a gravação inicie. **ATENÇÃO: esta opção apenas pode ser utilizada quando o canal de saída for do tipo Khomp. Caso este pré-requisito não seja cumprido, a gravação não será realizada.**
 - ♦ **R(<comando> <parâmetro1> <parâmetro2> ...)**: Executa <comando> ao final da gravação, como um comando de sistema operacional, passando como parâmetros <parâmetro1>, seguido de <parâmetro2>, ... (se existirem);
 - ♦ **s**: Grava em estéreo, sendo que um canal corresponderá ao envio de áudio para a placa, enquanto o outro corresponderá ao recebimento do áudio da placa;
 - ♦ **w**: Interrompe execução do *dialplan*, aguardando até o fim da ligação para continuar (útil em placas de gravação passiva);
- **outras-opções**: Define as opções de gravação a serem repassadas para o *application* **Monitor** ou **MixMonitor**, caso este seja chamado pelo **KRecord** (a escolha da aplicação depende da opção **s**: se estiver ajustada, a aplicação **Monitor** será usada; se não estiver, será utilizada a aplicação **MixMonitor**).

Exemplos:

```
exten => 1234,1,KRecord(teste.wav)
```

```
exten => 1234,1,KRecord(teste.wav|f)
```

Aplicação "KStopRecord"

Para a gravação de áudio iniciada anteriormente pela aplicação **KRecord**. A sintaxe da aplicação é a seguinte:


```
KStopRecord()
```

Esta aplicação não recebe parâmetros. Exemplo de utilização:

```
exten => 1234,1,KStopRecord()
```

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:

```
KSendSMS (recurso|destino[/opções] [|mensagem])
```

Podendo cada campo ser resumido em:

- **recurso:** Segue uma sintaxe idêntica à alocação de canais do application Dial, e define qual modem utilizar;
- **destino:** Número para onde enviar a mensagem;
- **opções:** Opções para setar parâmetros ou enviar SMS especiais:
 - ♦ **a:** Envia o SMS como mensagem de alerta. Também chamado de Flash SMS ou SMS de classe 0;
 - ♦ **c:** Requisita uma mensagem de confirmação de envio do SMS para a operadora;
 - ♦ **e(<encoding>):** Ajusta a codificação de envio do SMS, aonde **<encoding>** pode ser:
 - ♦ **7:** Envia a mensagem em codificação GSM 7 bits. Codificação padrão. As mensagens são sempre enviadas neste formato, a não ser que seja setada com uma das duas próximas opções;
 - ♦ **8:** Envia a mensagem em codificação GSM 8 bits;
 - ♦ **16:** Envia a mensagem em codificação GSM 16 bits. **Obs: Esta codificação só está disponível para envio de SMS do tipo WAP.**
 - ♦ **w(<URL>):** Requisita o envio de um SMS do tipo WAP PUSH com a URL **<URL>**. Este tipo de mensagem só pode ser enviada em codificação 8 ou 16 bits. São possíveis 2 tipos de WAP:
 - ♦ **Service Indication (SI):** É enviado um link para a URL e um texto na mensagem;
 - ♦ **Service Load (SL):** É enviado somente o link para a URL. Para enviar desta forma, é necessário deixar o campo **mensagem** vazio;
 - ♦ **p(<porta>):** Parâmetro **opcional** que define o número da porta de destino que a URL enviada no WAP será acessada. O valor padrão é a porta 2948.
- **mensagem:** Texto (sem aspas) que deve ser enviado para **destino**. Caso não seja fornecido o texto, a mensagem será enviada sem corpo.

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 channel.

Exemplos de uso desta aplicação seguem abaixo:

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

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(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:

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0|99887766|Mensagem de teste.)
exten => [...],n,NoOp(Enviou? ${KSmsDelivered})
exten => [...],n,NoOp(Codigo: ${KSmsErrorCode})
exten => [...],n,NoOp(Descr.: ${KSmsErrorName}) ; a partir do channel 3.0
```

- 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):

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0+b1|99887766|Mensagem de teste.)
```

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

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0|99887766/c|Mensagem de teste.)
```

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

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0|99887766/ac|Mensagem de teste.)
```

- Envia o WAP (SI) de link "www.teste.com" com o texto "Visite este site:" para telefone "99887766" utilizando o primeiro modem livre da placa "0", requisitando confirmação:

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0|99887766/e(8)cw(www.teste.com)|Visite este site:)
```

- Envia o WAP (SL) de link "www.teste.com" e porta de destino "1234" para telefone "99887766", utilizando o primeiro modem livre da placa "0":

```
exten => [...],1,NoOp(Enviando SMS...)
exten => [...],n,KSendSMS(b0|99887766/e(8)w(www.teste.com)p(1234))
```

Aplicação "KEchoCanceller"

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

Sintaxe atualizada da aplicação (**Disponível a partir da versão: 2.4.1**):

```
KEchoCanceller(ação[,opções])
```

Sintaxe anterior:

```
KEchoCanceller(ação)
```

Onde:

- **ação:** Pode ser **on** para habilitar o cancelador de eco, e **off** para desabilitar;
- **opções:** Define as opções do *application*, que são:
 - ♦ **N:** Executa a ação apenas no *channel* atual, não em toda a chamada.

Exemplo de uso desta aplicação:

```
exten => [...],n,KEchoCanceller(off)
```

Aplicação "KAutoGainControl"

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

Sintaxe atualizada da aplicação (**Disponível a partir da versão: 2.4.1**):

```
KAutoGainControl(ação[,opções])
```

Sintaxe anterior:

```
KAutoGainControl(ação)
```

Onde:

- **ação:** Pode ser **on** para habilitar o controle automático de ganho, e **off** para desabilitar;
- **opções:** Define as opções do *application*, que são:
 - ♦ **N:** Executa a ação apenas no *channel* atual, não em toda a chamada.

Exemplo de uso desta aplicação:

```
exten => [...],n,KAutoGainControl(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:

Sintaxe atualizada da aplicação (**Disponível a partir da versão: 2.4.1**):

```
KDTMFSuppression(ação[,opções])
```

Sintaxe anterior:

```
KDTMFSupression(ação)
```

Onde:

- **ação:** Pode ser **on** para habilitar a supressão DTMF, e **off** para desabilitar;
- **opções:** Define as opções do *application*, que são:
 - ♦ **N:** Executa a ação apenas no *channel* atual, não em toda a chamada.

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

Exemplo de uso desta aplicação:

```
exten => [...],n,KDTMFSuppression(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:

```
KSetVolume(<volume>)  
KSetVolume(<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:

```
KAdjustForFax()
```

Esta aplicação não recebe parâmetros. Exemplo de utilização:

```
exten => 1234,1,KAdjustForFax()
```

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:

```
KSendFax(<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:

```
exten => [...],n,KSendFax(/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:

```
KReceiveFax(<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:

```
exten => [...],n,Answer()  
exten => [...],n,KReceiveFax(/tmp/fax.tif)
```

Aplicação "KSelectSimCard"

Aplicação para seleccionar um SIM card em placas **KGSM**.

A sintaxe da aplicação é a seguinte:

```
KSelectSimCard([<placa>|<canal>|]<sim_card>)
```

Os campos têm o seguinte significado:

- **placa:** Define a qual placa enviar o comando.
- **canal:** Define a qual canal daquela placa enviar o comando.
- **sim_card:** Número do SIM card a ser seleccionado.

Exemplos:

```
exten => 1234,1,KSelectSimCard(0|2|1)
```

Aplicação "KSendTDD"

Esta aplicação tem a função de enviar texto utilizando o recurso **TDD (Disponível a partir da versão: 3.1)**, sendo a sua sintaxe a seguinte:

KSendTDD (<text>)

- **text:** Máximo 255 caracteres

Exemplo:

```
exten => [...], n, KSendTDD(Hello world!)
```

Aplicação "KReceiveTDD"

Esta aplicação tem a função de receber texto utilizando o recurso **TDD (Disponível a partir da versão: 3.1)**, sendo a sua sintaxe a seguinte:

KReceiveTDD ([timeout])

- **timeout:** Ajusta o tempo limite de espera dos caracteres, *default* igual a 5 segundos.

Canal "Khomp_SMS"

Este canal de comunicação é utilizado para receber mensagens SMS e criar ligações entrantes no Asterisk 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 Asterisk 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 Asterisk 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 Asterisk 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 **Monitor**, o **MixMonitor**, ou o **KRecord**), 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.

Interface de gerenciamento (AMI)

A interface de gerenciamento AMI permite a um programa externo controlar o Asterisk e as ligações gerenciadas pelo mesmo. Para facilitar este gerenciamento o channel da Khomp disponibiliza as seguintes facilidades:

Lista de comandos

Os comandos AMI permitem ao channel realizar funções auxiliares e/ou de ligações através de um programa externo. Os comandos disponibilizados são os seguintes:

KSendSMS

Envia mensagens SMS através de canais da placa KGSM da Khomp. Campos necessários para este comando:

- **Device:** especifica o dispositivo a alocar (para maiores informações, consultar a documentação do item **recurso** da aplicação **KSendSMS**);
- **Destination:** define o número de destino a enviar a mensagem;
- **Message:** contém a mensagem a ser enviada.
- **Confirmation:** quando setado como "true", requisita uma mensagem de confirmação de envio.

As respostas esperadas para este comando são duas: **Success** ou **Error**. Sendo a primeira indicando que a mensagem foi enviada com sucesso pelo modem, e a segunda indicando que houve erro no envio e detalhando o erro no campo **Message**.

Exemplos:

```
action: KSendSMS
device: b0c0
destination: 88888888
message: Teste
```

```
Response: Success
Message: Message sent
```

```
action: KSendSMS
device: b0
destination: 88888888
message: Teste.
```

```
Response: Error
Message: No free channel found
```


KHangup

Permite desligar chamadas forçando um envio de comando diretamente para a placa, utilizando índices de chamadas da placa KGSM da Khomp. Campos necessários para este comando:

- **Device:** especifica o dispositivo a desconectar (para o canal 0 da placa 0, por exemplo, utilizaria-se **B0C0**);
- **Index:** índice da chamada dentro do canal, utilizado em canais de placas **KGSM**.

Lista de eventos

Os eventos de AMI permitem ao channel comunicar eventos de ligações a um programa/aplicação externo que realiza o gerenciamento e/ou controle das ligações presentes no sistema. Os eventos disponibilizados são os seguintes:

Alarm

Reporta notificação de alarme em canal. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') onde ocorreu o alarme;
- **Alarm:** nome do alarme ocorrido no link (consultar capítulo "Códigos e significados").

AlarmClear

Reporta notificação de fim de alarme em canal. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') onde o alarme cessou.

AnswerInfo

Reporta detecção de atendimento no canal. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') onde ocorreu o alarme;
- **Info:** nome do tipo de atendimento detectado (para maiores informações, consulte a descrição detalhada da variável "KCallAnswerInfo").

AntennaLevel

Reporta alterações no nível de sinal da antena GSM; disponível apenas em placas KGSM. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') representando o modem;
- **Signal:** porcentagem do sinal, de 0 a 99 (inclusive).

OperatorRegistry

Reporta o registro bem sucedido do modem em uma operadora GSM. Disponível apenas em placas KGSM. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') representando o modem;
- **Operator:** nome da operadora.

BranchOffHook

Reporta que ramal está fora do gancho; disponível apenas em placas KFXS. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') que está fora do gancho.

BranchOnHook

Reporta que ramal está no gancho; disponível apenas em placas KFXS. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') que está no gancho.

CollectCall

Reporta detecção de chamada a cobrar. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') onde ocorreu o alarme.

NewSMS

Reporta que há uma nova mensagem SMS na placa; disponível apenas em placas KGSM. Contém as seguintes informações:

- **Channel:** nome do canal (em formato 'Khomp/BxCy') onde a mensagem foi recebida;
- **From:** número de telefone de onde a mensagem foi enviada (é fornecido pela operadora, e pode conter informação textual também);
- **Date:** data e hora do envio da mensagem SMS pela origem;
- **Size:** tamanho da mensagem (em bytes);
- **Mode:** codificação utilizada no envio da mensagem;
- **Message:** corpo da mensagem enviada.

OBS: Este evento só é enviado caso exista um contexto de recebimento de ligações SMS ajustado no dialplan; caso contrário, as mensagens SMS são mantidas no SIM card, evitando assim a perda destas mensagens.

KDisconnectionCause

Reporta a causa de desconexão de uma chamada em um canal Khomp.

- **Channel:** nome do canal (em formato 'Khomp/BxCy');
- **OrigAddr:** número de origem da chamada;
- **DestAddr:** número de destino da chamada;
- **Cause:** código numérico do desligamento;
- **Cause-txt:** causa do desligamento em texto.

Interface gateway (AGI)

Para facilitar a utilização do channel da Khomp, são fornecidos comandos AGI para determinadas funções, listados abaixo:

KSENDSMS

É um comando AGI para envio de mensagens SMS através do channel, utilizando codificação ISO-8859-1. O comando têm a seguinte sintaxe:

```
ksendsms <recurso> <destino> <mensagem>
```

Onde os campos têm o seguinte significado.

- **<recurso>** define o canal a ser alocado (para maiores informações, consultar o item "recurso" da documentação da aplicação "KSendSMS");
- **<destino>** é o telefone de destino, para onde a mensagem será enviada;
- **<mensagem>** contém o corpo da mensagem a ser enviada, sem aspas;

O código de retorno deste comando, em caso de sucesso, é:

```
200 result=1
```

Em caso de falha, o retorno é:

```
200 result=0 (<código-de-erro>)
```

Onde **<código-de-erro>** é um código definido pela API K3L, por exemplo:

```
200 result=0 (kgccUnallocatedNumber)
```

Caso o número de destino seja inexistente. Para maiores detalhes, favor consultar o capítulo seguinte, "Códigos e significados".

Multiparty em placas KGSM

A partir da versão **3.0**, o channel driver Khomp - aliado aos dispositivos **KGSM** - possui suporte ao recurso de **Multiparty**, permitindo conferência entre até cinco participantes remotos (dependendo da operadora), mais o canal de áudio da placa (participante local), assim como outras aplicações diversas que utilizem o recurso.

Pré-requisitos

Para utilizar o recurso de **Multiparty** das placas e dispositivos **KGSM**, é necessário:

- Ativar a configuração **Habilitar chamadas em espera** no **k3lconfig**, na seção de recursos GSM;
- Utilizar um *SIM card* cuja conferência de chamadas esteja ativa na operadora;
- Dispor de um nível de sinal de boa qualidade.

Conceitos básicos

O funcionamento das chamadas quando em modo Multiparty se dá a partir das seguintes regras:

1. Toda nova chamada entrante deve ser:
 - ♦ Atendida como uma chamada ativa, para que o áudio possa ser tratado por uma **URA** ou conferência;
 - ♦ Descartada sem ser atendida, para manter as chamadas atuais no estado que se encontram;
2. Não é possível manter mais de uma chamada ativa ou em espera sem que estas estejam em conferência;
3. Só pode haver uma conferência durante todo o percurso das chamadas.

Ou seja, as seguintes configurações são possíveis:

Número de		Cenários					
chamadas		1	2	3	4	5	6
Ativas		0	1	0	1	N	1
Em espera		0	0	1	1	1	N

Sendo **N** um número maior que "1" (um) e menor ou igual a "5" (cinco). A partir da segunda chamada recebida, pode-se colocar as chamadas já atendidas em conferência.

Exemplo:

- Considerando este cenário, na situação de atender uma terceira chamada, tendo uma chamada em espera e uma ativa: deve-se colocar todas as chamadas ativas em espera (unindo-as em conferência e depois colocando a conferência em espera), ou desconectar a chamada ativa. Após o atendimento, pode-se então

colocar todas as chamadas em conferência.

Chamadas entrantes

Contextos para chamadas adicionais

Ao receber uma nova chamada em uma placa KGSM, o *channel* tradicionalmente busca a mesma utilizando os contextos especificados no arquivo de configuração *khomp.conf*, conforme a documentação presente.

Entretanto, quando uma chamada ativa ou em espera já existe em um determinado canal, e uma nova chamada é recebida no mesmo, realiza-se uma primeira busca pelos contextos especificados para o canal KGSM em questão acrescidos do sufixo "-waiting".

Por exemplo, considerando o seguinte trecho de um arquivo de configuração:

```
context-gsm = khomp-gsm-DD  
context-gsm-alt = khomp-gsm
```

Uma chamada entrante pela placa 0 faria uma busca pelos contextos na seguinte ordem:

```
khomp-gsm-00  
khomp-gsm
```

No caso de uma segunda chamada entrante nesta mesma placa, com uma mesma configuração, em um canal que já possui uma chamada, o *channel* realiza uma busca na seguinte ordem:

```
khomp-gsm-00-waiting  
khomp-gsm-waiting  
khomp-gsm-00  
khomp-gsm
```

Isto permite a criação de contextos diferenciados para situações onde uma conferência já foi iniciada.

Extension waiting

Quando uma nova chamada é detectada pelo *channel*, e já existe uma chamada/conferência ativa sendo tratada pelo *dialplan*, não é possível atender esta nova chamada sem colocar a chamada atual em espera primeiramente.

Para manter este mecanismo sob o controle do programador do *dialplan*, quando uma nova chamada é detectada pelo *channel* nesta situação, realiza-se uma busca no contexto atual de execução do *dialplan* pelo *extension waiting*, efetuando um **Goto** para este *exten* caso ele exista.

Isto permite que seja programada uma ação específica (colocar a conferência em espera, por exemplo), permitindo tratar uma nova chamada de uma maneira controlada.

Para maiores detalhes, verificar o exemplo de *dialplan* ao final deste documento.

Novos *channels*

Khomp_MPTY

- Uso: representa logicamente uma conferência *multiparty*, executando em um contexto específico.

Um canal deste tipo é criado automaticamente quando alguma das chamadas executa o *application* **KGsmMultiparty**, sendo a chamada atual colocada em espera e o áudio redirecionado para o channel que representa a conferência.

Khomp_Wait

- Uso: representa logicamente uma chamada entrante que está em estado de espera.

Um canal deste tipo é criado automaticamente quando há um contexto de entrada definido na opção **context-gsm-wait** no arquivo de configuração, e uma nova chamada entrante chega num sistema onde já há uma chamada ativa em curso.

Este canal não possui nenhum tipo de tratamento de áudio ou de indicações, sendo executado apenas para permitir a execução de comandos externos ou o desligamento da chamada.

Pode-se forçar um desligamento da chamada atual executando a aplicação **Hangup** com o parâmetro **21** (*call reject*), da seguinte forma:

```
[khomp-waiting-00-01]
exten => s,1,Hangup(21)
```

O originante da chamada pode ser verificado através da variável **CALLERID(num)**.

Novos *applications*

KGsmMultipartyStart

- Uso: invocado a partir de um chamada que deseja iniciar uma conferência, ou a partir de um canal de conferência já criado, para agrupar todos os canais atuais numa mesma conferência.

Recebe, como parâmetro:

1. um contexto (opcional);
2. um extension (opcional);

3. uma prioridade (opcional).

A tripla contexto/*extension*/prioridade o início da execução do plano de discagem relativo à conferência. O *extension* padrão, caso omitido, é "s", e a prioridade padrão é 1. Se o canal **Khomp_MPTY** já foi inicializado por outra ligação anteriormente, o seu fluxo de execução será retomado, e desviado para a tripla contexto/*extension*/prioridade. Se nenhum parâmetro for informado, a execução será retomada de onde foi interrompida.

IMPORTANTE: Caso não seja especificado um contexto em nenhuma invocação do *application* dentro da chamada atual, a criação do canal **Khomp_MPTY** não será realizada, não sendo possível realizar nenhum controle de fluxo para a conferência.

Exemplos:

```
; atende uma ligação e cria um canal de conferência.  
[mpty]  
exten => s,1,WaitForSilence(99999)  
  
[khomp-gsm]  
exten => s,1,Answer()  
exten => s,n,KGsmMultipartyStart(mpty)
```

KGsmMultipartyStart2

- **Uso:** funciona da mesma forma que o *application* **KGsmMultipartyStart**, mas permite a especificação de opções de execução adicionais.

Recebe, como parâmetro:

1. opções de execução (opcional);
2. um contexto (opcional);
3. um *extension* (opcional);
4. uma prioridade (opcional).

As seguintes opções de execução podem ser utilizadas:

- **f:** Determina que o primeiro canal a entrar no **Multiparty** deverá ser o dono da conferência - assim que este canal desligar, todos outros canais também são desconectados.
IMPORTANTE: Todos participantes que devem ser associados ao dono da conferência precisam necessariamente ser colocados nesta utilizando o *application* **KGsmMultiparty** com o parâmetro **f** ajustado. Os participantes que não receberem este parâmetro não serão desassociados da conferência com o desligamento do dono.
- **G:** Se o canal de conferência já foi criado, ignora os parâmetros contexto/*extension*/prioridade e apenas retorna a execução do canal de conferência **Khomp_MPTY** para o ponto de onde foi interrompido;
- **X(n):** Seguida de um parâmetro numérico **n**, exclui o participante **n** da conferência, mantendo-o em espera.

O processamento dos outros parâmetros segue a mesma lógica da aplicação **KGsmMultipartyStart**.

Exemplos:

```
; atende uma ligação e cria um canal de conferência.  
[mpty]  
exten => s,1,WaitForSilence(99999)  
  
[khomp-gsm]  
exten => s,1,Answer()  
exten => s,n,KGsmMultipartyStart2(f,mpty)
```

KGsmMultiparty

- Uso: invocado a partir de um chamada que deseja iniciar uma conferência, ou a partir de um canal de conferência já criado, para agrupar todos os canais atuais numa mesma conferência.

Recebe, como parâmetro:

1. um contexto (opcional);
2. um extension (opcional);
3. opções de execução (opcional).

Este *application* está obsoleto, favor utilizar o *application* **KGsmMultipartyStart** em seu lugar.

KGsmMultipartyBreak

- Uso: invocado a partir de uma conferência, quebra a conferência atual colocando o dono desta novamente em execução, e os outros participantes em espera - interrompendo a execução do canal de conferência **Khomp_MPTY**, se este existir. A execução deste canal retorna quando o dono da conferência executar o *application* **KGsmMultiparty**.

Por padrão, o *application* faz com que o canal relativo ao dono da conferência tenha sua execução retomada - ou seja, a execução atual do *application* **KGsmMultiparty** pelo dono da conferência retornará.

Entretanto, pode-se especificar uma nova localização do *dialplan* para onde esta execução deve prosseguir, com os seguintes parâmetros:

1. um contexto (opcional);
2. um *extension* (opcional);
3. uma prioridade (opcional).

Exemplos:


```
; quebra a execução da conferência caso o usuário digite "#".
[empty]
exten => s,1,WaitExten(999)
exten => s,n,Goto(1)

exten => #,1,KGsmMultipartyBreak()
exten => #,n,Goto(s,1)

; quebra a execução da conferência caso o usuário digite "#"
; e redireciona o dono para o contexto "break", exten "s",
; prioridade "1" (mesma sintaxe do comando Goto).
[empty]
exten => s,1,WaitExten(999)
exten => s,n,Goto(1)

exten => #,1,KGsmMultipartyBreak(break,s,1)
exten => #,n,Goto(s,1)
```

KGsmMultipartyOwner

Invocado a partir de uma chamada ou conferência, ajusta o dono da conferência para um novo valor, ou desabilita o recurso completamente.

Recebe, como parâmetro:

1. o número do novo dono.

Como número do novo dono, podem ser utilizadas as palavras reservadas "none", para nenhum dono, ou "detach" para desabilitar o recurso de dono da conferência.

Exemplos:

```
; ajustando o dono da conferência para o segundo participante (número 1)
exten => s,1,KGsmMultipartyOwner(1)

; desabilitando temporariamente o dono da conferência nesta chamada.
exten => s,1,KGsmMultipartyOwner(none)
```

KGsmDial

- Uso: invocado a partir de uma chamada ou conferência ativa, visando realizar uma chamada para um número externo utilizando o mesmo *chip* da chamada atual.

Recebe, como parâmetros:

1. o número a discar;

2. opções de discagem (opcional);
3. um contexto (opcional);
4. um *extension* (opcional).

A chamada/conferência atual fica retida até o número ser desconectado, ou até ser atendido e colocado em espera (invocar o *application* **KGsmHold**), ou até ser atendido e colocado em conferência (invocar o *application* **KGsmMultiparty**).

Como opções de discagem, os seguintes modificadores são aceitos:

- **f**: Quando os parâmetros contexto e *extension* não são fornecidos, uma nova chamada sempre entra em conferência ao ser atendida (através do *application* **KGsmMultiparty**). Caso o parâmetro **f** seja repassado para o *application* **KGsmDial**, ela será repassada para o *application* **KGsmMultiparty**, fazendo com que a nova chamada tenha como dono o primeiro participante da conferência;
- **M**: Apenas retorna quando o canal original que invocou o *application* voltar a ser o canal ativo. Se esta opção não for especificada, o *application* também retorna se o canal de conferência se tornar o canal ativo;
- **o**: Caso a ligação seja atendida com sucesso, esta passará a ser a dona da conferência;
- **T(timeout)**: Ajusta *timeout* como tempo máximo para estabelecimento da chamada.

Como retorno da ligação, duas variáveis são ajustadas:

- **KGsmDialStatus**
Possui os valores "**SUCCESS**", "**FAILURE**", "**ABORTED**" e "**TIMEOUT**", indicando - respectivamente - se a chamada completou com sucesso, se alguma falha foi detectada pela operadora ou pelo modem, se a chamada foi abortada, ou se ocorreu um *timeout*.
- **KGsmDialCode**
Possui o código de erro específico da chamada, e é ajustado quando o resultado da variável **KGsmDialStatus** for igual a "**FAILURE**". Os valores possíveis podem ser consultados neste documento, na seção Códigos de chamada.

Exemplos:

```
; chamando número 99887766 e colocando-o em conferência
; logo após o atendimento.
[mpty]
exten => s,1,KGsmDial(99887766)

; chamando número 99887766 e colocando-o em conferência
; logo após o atendimento, com recurso de "dono" ativado.
[mpty]
exten => s,1,KGsmDial(99887766,f)
```

KGsmHold

- Uso: invocado a partir de uma chamada ou conferência, coloca a chamada ou conferência ativa em espera, atendendo a chamada entrante ainda não atendida (se houver) ou re-iniciando a execução de uma chamada ou conferência anteriormente colocada em espera. Caso não exista nenhuma chamada entrante aguardando para ser atendida, ou nenhuma outra chamada/conferência em espera, o *application* não realiza ação alguma.

Este *application* não recebe parâmetros.

Novos comandos

khomp calls show

- Uso: mostra os canais GSM e suas chamadas associadas.

Recebe, como parâmetro, um número de dispositivo, que é utilizado para limitar a listagem dos canais pertencentes a este dispositivo.

Acréscimos no AMI

Novos comandos

KHangup

Desliga uma chamada, enviando diretamente um comando para a placa.

- Campos:
 - ♦ Channel: Nome do canal (Khomp/BnCy), sendo **x** a placa, **y** o canal.
 - ♦ Index: Índice da chamada.

Novos eventos

HoldStart

Indica que uma determinada chamada está em espera.

- Campos:
 - ♦ Channel: Nome do canal (Khomp/BnCy-z) onde ocorreu o evento, sendo **x** a placa, **y** o canal e **z** o

índice da chamada.

HoldStop

Indica que uma determinada chamada não está mais em espera.

- Campos:
 - ♦ Channel: Nome do canal (Khomp/BnCy-z) onde ocorreu o evento, sendo **x** a placa, **y** o canal e **z** o índice da chamada.

MptyStart

Indica que uma determinada chamada está em conferência.

- Campos:
 - ♦ Channel: Nome do canal (Khomp/BnCy-z) onde ocorreu o evento, sendo **x** a placa, **y** o canal e **z** o índice da chamada.

MptyStop

Indica que uma determinada chamada não está mais em conferência.

- Campos:
 - ♦ Channel: Nome do canal (Khomp/BnCy-z) onde ocorreu o evento, sendo **x** a placa, **y** o canal e **z** o índice da chamada.

Exemplos de *dialplan*

Seguem abaixo alguns exemplos de *dialplan* que realizam conferência com os recursos *Multiparty* do channel.

Conferência simples

Este exemplo aguarda chamadas entrantes, e coloca-as em conferência, tocando mensagens de aviso para os usuários conforme novas ligações são recebidas.

- khomp.conf

```
context-gsm-call = khomp-gsm
```

- extensions.conf

```
[default]
exten => s,1,Playback(fpm-calm-river)
exten => s,n,Goto(1)
```

```
exten => waiting,1,NoOp(waiting call)
exten => waiting,n,Playback(conf-lockednow)
exten => waiting,n,KGsmHold()
exten => waiting,n,Playback(conf-unlockednow)
exten => waiting,n,Goto(${WAITINGEXTEN}|1)
```

```
[khomp-gsm]
exten => s,1,NoOp(first call)
exten => s,n,Answer()
exten => s,n,Wait(2)
exten => s,n,Playback(hello-world)
exten => s,n,KGsmMultiparty(default)
exten => s,n,Playback(vm-goodbye)
```

```
[khomp-gsm-waiting]
exten => s,1,NoOp(other calls)
exten => s,n,Answer()
exten => s,n,Wait(2)
exten => s,n,Playback(conf-enteringno)
exten => s,n,Playback(digits/0)
exten => s,n,KGsmMultiparty()
```

Conferência avançada

Este exemplo aguarda chamadas entrantes, e colocando-as em conferência, permitindo ao dono da mesma (o primeiro participante) realizar diversas ações - discar para um novo participante e colocá-lo numa conferência, conversar em particular com o mesmo^[1], transferir a propriedade da conferência para outro participante, ou até mesmo excluir-se da conferência e permitir que a mesma continue até o último participante desligar.

1. ² Desde que obedecidas as restrições do protocolo GSM, conforme tabela de estados válidos listada no início deste capítulo.

```
;;[general]
;;autofallthrough=no
;;clearglobalvars=yes

;; Context for default calls (incoming).
[khomp-gsm]
exten => s,1,Set(GLOBAL(PHONE_${CHANNEL:6:4})=00000000)

;; WARNING: If using Local, always leave a playback here (even if just 1ms of silence).
;; WARNING: This is *REQUIRED* to Local optimize itself out of the way and allow KGsm*
;; WARNING: applications to access the underlying Khomp channel.
exten => s,n,Playback(hello-world)

exten => s,n(mpty),KGsmMultiparty(khomp-mpty|s|f)
exten => s,n(disa),DISA(no-password|disa-mpty)
exten => s,n,Goto(disa)
exten => s,n,Hangup()

;; Context for dialed calls using KGsmDial, in private mode.
[khomp-gsm-private]
exten => s,1,KGsmMultiparty(||fX(1))
exten => s,n,Goto(khomp-mpty,s,wait)

;; Context for reading digits.
[disa-mpty]
```

```
exten => _1X.,1,Set (GLOBAL (PHONE_${CHANNEL:6:4})=${EXTEN})
exten => _1X.,n,Goto (khomp-gsm|s|empty)

exten => _[23456789]XXXXXXX,1,Set (GLOBAL (PHONE_${CHANNEL:6:4})=${EXTEN})
exten => _[23456789]XXXXXXX,n,Goto (khomp-gsm|s|empty)

exten => _0XX[23456789]XXXXXXX,1,Set (GLOBAL (PHONE_${CHANNEL:6:4})=${EXTEN})
exten => _0XX[23456789]XXXXXXX,n,Goto (khomp-gsm|s|empty)

exten => *,1,Set (GLOBAL (PHONE_${CHANNEL:6:4})=${EXTEN})
exten => *,n,Goto (khomp-gsm|s|empty)

exten => i,1,Playback (beeperr)
exten => i,n,Goto (khomp-gsm|s|disa)

exten => t,1,Playback (beeperr)
exten => t,n,Goto (khomp-gsm|s|disa)

exten => T,1,Playback (beeperr)
exten => T,n,Goto (khomp-gsm|s|disa)

;; Context where Khomp_MPTY channel runs.
[khomp-mpty]
exten => s,1,NoOp()
exten => s,n(wait),WaitExten()
exten => s,n,Goto (wait)
exten => s,n,Hangup()

exten => i,1,Playback (beeperr)
exten => i,n,Goto (s|wait)

exten => t,1,Playback (beeperr)
exten => t,n,Goto (s|wait)

exten => T,1,Playback (beeperr)
exten => T,n,Goto (s|wait)

exten => *1,1,KGsmMultipartyBreak()
exten => *1,n,Set (PHONE=${GLOBAL (PHONE_${CHANNEL:11:4})})
exten => *1,n,Gosub (mpty-action,s,1)
exten => *1,n,Goto (s|wait)

exten => *2,1,KGsmMultipartyBreak()
exten => *2,n,Set (PHONE=${GLOBAL (PHONE_${CHANNEL:11:4})})
exten => *2,n,Gosub (mpty-private,s,1)
exten => *2,n,Goto (s|wait)

;; Multiparty context for private dialed calls.
[mpty-private]
exten => s,1,NoOp (GOT PHONE ${PHONE})
exten => s,n,KGsmDial (${PHONE}|of|khomp-gsm-private)
exten => s,n,WaitExten (9999)

exten => #1,1,KGsmMultiparty (||f)
exten => #1,n,Return ()

exten => #2,1,KGsmMultipartyOwner (2)
exten => #2,n,KGsmMultiparty (||f)
exten => #2,n,Return ()
```

```
;; Multiparty context for ordinary dialed calls.
[empty-action]
exten => s,1,NoOp(GOT PHONE ${PHONE})
exten => s,n,Goto(${PHONE},1)

exten => _XX.,1,KGsmDial(${PHONE}|of|khomp-gsm)
exten => _XX.,n,Return

exten => *,1,KGsmMultipartyOwner(none)
exten => *,n(play),Playback(vm-goodbye)
exten => *,n,Wait(15)
exten => *,n,Goto(play)
```

Compatibilidade com estilo DAHDI de Dialstring

(Disponível a partir da versão: 3.1
)

A configuração do DAHDI é agrupada por *Spans* que podem ser placas/links. Esses *Spans* servem como agrupamentos lógicos de canais e podem ser configurados nos grupos manualmente ou de forma automática. Os canais de todas as placas são configurados/apresentados em uma única sequência começando pelo 1 e sem a distinção de placa.

Comportamento compátivel com DAHDI

A compatibilidade com o DAHDI é realizada da seguinte forma:

- Aplicação Dial:
 - ◆ Dial(Khomp/<channel#>[clr<cadance#>ld]/<extension>)
 - ◆ Dial(Khomp/(glGlrR)<group#(0-63)>[clr<cadance#>ld]/<extension>)
 - ◆ Dial(Khomp/i<span#>/<extension>)
- Opções de grupos:
 - ◆ g - alocação de grupos de canais com busca ascendente
 - ◆ G - alocação de grupos de canais com busca descendente
 - ◆ r - alocação de grupos de canais com busca *round robin* ascendente
 - ◆ R - alocação de grupos de canais com busca *round robin* descendente
- Opções gerais:
 - ◆ i - ISDN restrição de canal por *Span*. (**aceita, mas não utilizado**)
 - ◆ c - espera por dígito DTMF para confirmar o atendimento (**aceita, mas não utilizado**)
 - ◆ r<cadance#> - configura cadência da chamada
 - ◆ d - força capacidade de suporte para ISDN/SS7 em chamada (**aceita, mas não utilizado**)
- Para ativar a compatibilidade com o DAHDI é necessário setar a opção **dial-string-like-dahdi** do arquivo de configuração **khomp.conf** para **yes**. Além disso, essa configuração modifica a visualização de alguns comandos do channel da Khomp.
- Os grupos funcionam como os agrupamentos lógicos do DAHDI, sendo os grupos padrões do DAHDI configurados automaticamente.
- As cadências no DAHDI são indicadas por números no **khomp.conf**.

Exemplos de Dialstrings

```
Dial (Khomp/1/2927)
Dial (Khomp/2r1/2927)
Dial (Khomp/G1/32625644)
Dial (Khomp/R1r2/99991234)
```


Dial (Khomp/i2/30110666)

Utilização de *patches* adicionais

Para alguns recursos disponibilizados pelo channel - como no caso da detecção e reportação de atendimento por caixa postal e secretária eletrônica - pode ser necessário o uso de *patches* para o Asterisk®, que visam estender as capacidades deste *software* para atender melhor às necessidades do channel da Khomp.

Estes *patches* encontram-se na distribuição do código-fonte do channel da Khomp, em um diretório denominado **patches**, onde cada diretório contém patches para uma ou mais versões do Asterisk®, e arquivos de documentação explicando suas funcionalidades.

Para aplicar estes patches, deve-se seguir estes procedimentos básicos:

- Descompactar o código-fonte do Asterisk:

```
$ tar -zxvf asterisk-1.x.y.tar.gz
```

- Alternar o diretório corrente para o diretório base do código-fonte do Asterisk:

```
$ cd asterisk-1.x.y
```

- Executar o comando **patch**, repassando o *patch* em questão como base:

```
$ patch -p0 < patch-xyzw.diff
```

- Recompilar o Asterisk e, caso indicado na documentação, o channel da Khomp também.

Códigos e significados

Este capítulo apresenta os códigos presentes no channel da Khomp e seus significados, utilizados tanto nos eventos AMI quanto 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;
- **SMS In Progress:** o canal está ocupado enviando/recebendo mensagens SMS;
- **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 (Asterisk)

Reflete diretamente o estado de chamada controlado pelo Asterisk, que pode ser:

- **down:** Canal desconectado;
- **reserved:** Canal alocado para posterior ligação sainte;
- **offhook:** Canal alocado para ligação sainte;
- **dialing:** Canal em processo de discagem;
- **ring:** Ligação entrante chamando;
- **ringing:** Ligação sainte chamando;
- **ongoing:** Ligação estabelecida;
- **busy:** Ligação sainte para canal ocupado;
- **offdial:** Não utilizado pelo canal da Khomp (*offhook dialing*);
- **mute:** Não utilizado pelo canal da Khomp (*mute*).

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 *channel* 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 Asterisk 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 Asterisk não iniciará.**

Se você deseja rodar o sistema sem a placa da Khomp, é preciso configurar o asterisk para ele não carregar o módulo da Khomp. Para isso, abra o arquivo `/etc/asterisk/modules.conf`, e adicione a linha:

```
noload => chan_khomp.so
```

Quando a placa da Khomp estiver devidamente configurada e os módulos da khomp carregados (explicado acima), lembre-se de retirar esta linha do arquivo.

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

```
# kserver start
```

Após realizar estes procedimentos, o channel já estará operacional, e o Asterisk 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 channel**, 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 channel 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;
/etc/asterisk/    -- Configurações do Asterisk e channel;
/usr/doc/khomp/   -- Documentação das placas, do chan_khomp e dos utilitários;
/usr/sbin/        -- Utilitários e servidor de processos;
/usr/lib/         -- Bibliotecas compartilhadas da K3L;
/usr/lib/asterisk/modules/ -- Módulo 'chan_khomp.so';
/var/log/khomp2.1/ -- Diretório de logs da K3L e channel
                  (channel 3.0);
/var/log/khomp2.0/ -- Diretório de logs da K3L e channel
                  (channel 2.4);
```

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](#).

Compatibilidade com módulo Zaptel/DAHDI

Em alguns casos, quando a instalação do Asterisk é feita por algum gerenciador de pacotes, os *drivers* Zaptel/DADHI podem ser carregados por padrão - mesmo não havendo necessidade. Caso estes não sejam utilizados, sugere-se que seja removidos visto questão de compatibilidade - explanadas com maior detalhamento a seguir.

Seguem os passos para verificar se o módulo está carregado e como removê-lo:

- Para saber se os módulos estão carregado:

```
# lsmod | grep zaptel
# lsmod | grep dahdi
```

- Caso algum dos comando acima retornar algum módulo, significa que pelo menos um dos *drivers* está carregado. Para removê-lo, execute:

```
# rmmmod zaptel
```

ou

```
# rmmmod dahdi
```

- É necessário certificar que esse módulo não seja novamente carregado. No Debian isso pode ser feito removendo o pacote **zaptel**, ou o pacote **dahdi**:

```
# apt-get remove zaptel
```

ou

```
# apt-get remove dahdi
```

Caso necessário, é possível utilizar os módulos Zaptel/DAHDI com as placas da Khomp, mas para isso é necessário inicializar **primeiramente** os módulos da Khomp.

Isto se deve a uma incompatibilidade entre o sistema de detecção de dispositivos implementada nos *drivers* Zaptel/DAHDI com as placas Khomp - que são incorretamente inicializadas por estes - e impede a sua correta detecção pelos *drivers* da Khomp.