

Guia de Referência do Quotemod 2 v1.2

Controle de versões do documento

Data	Descrição	Versão	Atualizado por
10/09/2009	Criação do documento	1.0	Wagner Alves
14/09/2009	Adição dos itens convenções e conexão	1.1	Wagner Alves
21/09/2009	Adição de informações sobre os campos	1.2	Wagner Alves
22/10/2010	Alteração comando Logon Inclusão de chamadas Advise de serviços	1.3	Claudiomar Moreira

1 – Introdução

O Quotemod é um roteador de comandos para os servidores de conteúdo da Agência Estado. Esses servidores disponibilizam uma interface de comando cujo acesso é feito através do Quotemod.

O Quotemod transforma o protocolo Broadcast (proprietário da Agência Estado) em um protocolo texto simplificado, para isso, possui uma interface nativa de comandos que será descrita no próximo item. Dentre as funcionalidades do Quotemod podemos destacar:

- Disponibiliza uma interface simples para executar comandos e obter dados do servidor de informações da Agência Estado;
- Cada sessão de usuário é vinculada a uma conexão física (socket) com o Quotemod;
- Existe apenas uma porta de conexão com o Quotemod;
- O usuário é autenticado junto ao servidor, não sendo possível autenticar mais de uma sessão ao mesmo tempo para o mesmo usuário;
- O Quotemod envia um keep-alive ao usuário, caso nenhuma outra requisição esteja sendo atendida, contendo data/hora do servidor.

2 – Convenções

\n - caractere de nova linha.

\t - caractere de tabulação.

3 – Conexão

O endereço IP do Quotemod é passado pela Produção da Agência Estado e a porta de conexão é a **8081**. Tanto as solicitações como as respostas do servidor são trafegadas pela mesma conexão.

4 – Interface de Comandos

Todos os comandos seguem o padrão abaixo:

```
<ID>\t<COMANDO>\t<SERVICO>\t<PARAM1>\t...<PARAMn>\n
```

Descrição:

- \t – Caractere TAB;
- \n – Quebra de Linha;
- <ID> – Número de identificação do cliente. Será devolvido juntamente com a resposta do servidor;
- <COMANDO> – Comandos nativos do Quotemod. São suportados os seguintes comandos:
 - START – Inicia um advise;
 - STOP – Interrompe um advise;
 - EXEC – Executa uma função no **servidor de conteúdo**, as funções do servidor serão demonstradas adiante;
 - REQ – Executa uma consulta a um conteúdo estático do servidor;
 - POKE – Insere um dado no servidor;
 - LOGON – Inicia uma sessão de usuário no servidor;
 - LOGOFF – Fecha uma sessão de usuário no servidor;
 - KEEP – Evita que sessão seja fechada automaticamente por timeout. Deve ser emitido periodicamente caso nenhuma outra transação seja solicitada;
- <SERVICO> – Serviço e tópico a ser consultado. Ex.: BC|ATIVO, API|API, etc.
- <PARAMn> – Parâmetro pertinente a solicitação desejada. Ex.: PETR4, VALE5, etc.

5 – Respostas

O sistema gera diferentes tipos de respostas conforme a consulta ou comando executado.

Resposta para Comando

```
C\t<ID>\t<TAMANHO>\t<DADOS>
```

Onde:

- C – Indica que se trata de uma resposta de comando
- \t – Caractere TAB (#9)
- <ID> – Número de identificação do cliente. Foi enviado juntamente com a requisição
- <TAMANHO> – Tamanho (em bytes) do conteúdo retornado pelo servidor
- <DADOS> – Dados retornados pelo servidor (array de bytes).

Respostas de Erro**E\t<ID>\t<TAMANHO>\t<Mensagem de erro>**

Onde:

E – Indica que se trata de uma mensagem de erro**\t** – Caractere TAB (#9)**<ID>** – Número de identificação do cliente. Foi enviado juntamente com a requisição**<TAMANHO>** – Tamanho (em bytes) do conteúdo retornado pelo servidor**<Mensagem de erro>** – Mensagem de erro enviado pelo servidor.**Advices**

Advices são notificações automáticas de alteração de um determinado item. Através desse mecanismo, um usuário pode ser avisado sobre alterações no valor de um papel, atualização de hora no servidor, avisos de mudança nas permissões do usuário, etc. Para começar a receber essas notificações para um item, o mesmo deve ser assinado através do comando START.

Advices são enviados pelo Quotemod com a seguinte estrutura:

A\t<SERVICO|TOPICO>\t<ITEM>\t<TAMANHO>\t<DADOS>

Onde:

A – é o ID que indica que se trata de um advise;**\t** – Caractere TAB (#9);**<SERVICO|TOPICO>** – Serviço e tópico que gerou a notificação. Ex.: BC|ATIVO, API|API, etc.;**<ITEM>** – Item notificado. Ex.: PETR4, VALE5, MANCHETES etc.**<TAMANHO>** – Tamanho (em bytes) do conteúdo retornado pelo servidor**<DADOS>** – Dados retornados pelo servidor

Exemplo:

Comando: Id\tSTART\tBC|ATIVO\tpetr4

Resposta consulta: C\tid\t3\tOK\0

Resposta advise: A\tBC|ATIVO\tPETR4\t300 \tPETR4\t 33.05\t 17:07\t...\0

O conjunto SERVICO|TOPICO e ITEM é único dentro do sistema. No exemplo acima, o cliente deve saber que em um momento foi solicitada a informação sobre o serviço “BC|ATIVO” com o parâmetro “PETR4”.

6 – Comandos

Para a exemplificação dos comandos usaremos 1 como ID.

Observação: Para consultas que retornam sucesso ou não sucesso será adotado como padrão de resposta: "OK" – sucesso e "NOK" – para não sucesso, no campo <dados>, lembrando este campo é um array de bytes e portando para strings o caractere \0 será considerado. Alguns comandos onde são passados mais de um parâmetro a resposta negativa poderá ser seguida dos parâmetros negados, separados por TAB(\t). Para exemplos veja a tabela 2. O formato dos dados retornado em <dados> pode variar conforme o comando solicitado.

Tabela 1 - Exemplos de comandos X resposta

Comando	Respostas
Id\tlogon\tuser\tpsw\n	C\tid\t3\tOK\0 ou se ocorrer erro E\tidt\28\tUsuários e/ou senha inválidos\0 Obs. O tamanho é 3, pois o array de bytes contém o caractere \0.
Id\tStart\tBC ATIVO\t*\n	C\tid\t3\tOK\0
Id\tSTART\tBC ATIVO\tpetr4\t ppp\n	C\tid\t8\tNOK\tppp\0
Id\tREQ\t tBC ATIVO \tsss\n	C\tid\t4\tNOK\0
Id\tEXEC\tAPI API\tPegaNoticias(BROADCAST, AE-News, *, 260609)\n	C\tid\t4\tNOK=Data invalida. \0 <DADOS>= NOK=Data invalida. \0

6.1 – Comando LOGON

Faz o logon de um usuário previamente cadastrado no sistema.

Sintaxe

Comando: id\tLogon\tusuário\tsenha[\t<lista de parâmetros>]\n

Parâmetros do comando

Usuário – Nome do usuário que será autenticado

Senha – Senha do usuário a ser autenticado

lista de parâmetros (*opcional*): lista de parâmetros para a sessão no

formato<chave>=valor;...;<chave_N>=<valor_N> Os parâmetros atualmente reconhecidos são:

- Software: id do software na BCAA (ex: **27** para "**AE MOVEL BROKER**"). Se não for declarado, será assumido o valor default configurado no servidor AEDataFeed
- Language: id do idioma a ser utilizado nas respostas de erro. Default = "**pt**"
- UserGroup: grupo a que pertence o usuário. Default = "**1**" (grupo "AE" - atualmente não existem outros grupos definidos)
- TokenDisconnect: token de sessão da BCAA a ser desconectada, caso necessário (ver observação abaixo)

Resposta

Sucesso: C\t<Id>\t<Tamanho>\t<Status>

Erro: E\t<Id>\t<Tamanho>\t<Erro>

Exemplo de Aplicação

Comando: 1\tLogon\tusuário\tsenha\n

Resposta: C\t1\t3\tOK\0 ou E\tId\t28\tUsuários e/ou senha inválidos

Obs:

Em caso de erro, o aplicativo pode verificar se o motivo foi a presença de sessões BCAA para o mesmo login em outras plataformas, e, nesse caso, mostrar uma lista dessas sessões para que o usuário escolha qual delas deve ser desconectada. Uma vez escolhida uma sessão, deve ser tentado novo Logon passando o id da sessão a desconectar via parâmetro **TokenDisconnect**. Caso haja sessões a desconectar, o campo <Erro> terá o seguinte formato:

- **OpenSessions\t<lista de sessões>** onde
 - **<lista de sessões>** é a lista de sessões atualmente ativas para o login, no formato **<token>;<plataforma>;<data>**\n...

6.2 – Comando LOGOFF

Termina a sessão e desconecta o usuário.

Sintaxe

`id\tLogoff\n`

Parâmetros do comando

Não se aplicam

Resposta

`C\tId\tTamanho\tStatus`

Exemplo de Aplicação

Comando: `1\tLogoff\n`

Resposta: `C\t1\t3\tOK\0`

6.3 – Comando KEEP

Keep alive do socket. Deverá ser enviada apenas caso nenhum outro comando seja enviado num determinado período.

Sintaxe

`id\tKEEP\n`

Parâmetros do comando

Não se aplicam.

Resposta

Não se aplica

Exemplo de aplicação

`1\tKEEP\n`

6.4 – Comando EXEC

O comando EXEC do Quotemod executa funções dos servidores de conteúdo. Essas funções são acessíveis somente através do comando EXEC.

Sintaxe

```
id\tEXEC\tSERVICO|TOPICO\tFUNÇÃO(PAR1, PARn)\n
```

Parâmetros do comando

Não se aplicam.

Resposta do comando

Descrito na tabela de funções.

Exemplo de aplicação

A tabela abaixo mostra algumas funções mais comumente usadas.

Função	Parâmetros	Descrição	Exemplo de aplicação
OpenUserSession	<p>usuario: login do usuário</p> <p>senha: [senha] ou ?+[hash multicript] do usuário, para autenticação</p> <p>software: id do software, de acordo com a tabela Identificadores da BCAA</p> <p>grupo (opcional): grupo a que pertence o usuário - se vazio, será assumido default do servidor</p> <p>idioma (opcional): idioma em que a BCAA deve enviar mensagem de erro - se vazio, será assumido default do servidor</p> <p>token_desconecta (opcional): token de sessão da BCAA que deve ser desconectada para atender este pedido de logon</p>	<p>Abre uma sessão de usuário. O usuário pode pertencer à base local ou à BCAA. Este método pode ser utilizado para abrir uma sessão e retornar a lista de serviços permissionados para o usuário. O aplicativo cliente é responsável pelo tratamento das permissões retornadas.</p>	<p>Comando:</p> <pre>1\tEXEC\tAEUSER PERM\OpenUserSession(usuario, senha, software, idioma="pt", token_desconecta="")\n</pre> <p>Resposta:</p> <p>Sucesso:</p> <pre>OK=<produto>=<serviço>\n... \n<serviço_N>\r ... \r<produto_N>=<serviço>\n... \n<serviço_N></pre> <p>Erro (dependendo da fonte da mensagem):</p> <pre>NOK=<Mensagem de erro> ou NOK=<código de erro>\t<mensagem de erro></pre>
PegaDatas	Não se aplicam	Retorna todas as datas de notícias disponíveis no servidor.	<p>Um só comando:</p> <p>Comando: id\tEXEC\tAPI API\tPegaDatas()\n</p> <p>Resposta: C\tid\tbytes\tOK=DDMMAA\tDDMMAA</p>
PegaServicosCategorias	<p>Produto – BROADCAST</p> <p>Serviço – nome do serviço. Pode ser usado</p>	Retorna todos os cadernos permissionados para o usuário ().	<p>Comando:</p> <pre>1\tEXEC\tAPI API\tPegaServicosCategorias(BROADCAST,*) \n</pre>

	o * para todos.		Resposta: C:\t1\t2555\tOK= *servicosAE\n*Ptr=par. Soja Grão\t...\n
PegaNoticias	Produto -BROADCAST Caderno - resultado da função anterior. Categoria - * para todos. Data - Data, no formato DDMMAA.	Retorna todas as noticias de um caderno	Comando: 1\tEXEC\tAPI API\tPEGANOTICIAS(BROADCAST, AE-News, *, 040509)\n Resposta: C\t1\t23847\tOK=040509;AE-News;AE- News;N008B090504; Ao vivo: Álvaro Bandeira, Da Ágora, Fala...

Função	Parâmetros	Descrição	Exemplo de aplicação
SolicitaTexto	ID - Identificador único da noticia, recebido na função "peganoticias".	Retorna o conteúdo de uma noticia	Comando: 1\tEXEC\tAPI API\tSolicitaTexto(N008B090504)\n Resposta: C\t1\t2254\t<HTML><BODY bgcolor=white>\n<dd> 14:22 G-7: ... </HTML>corpo\tConteúdo
GetPermUser	Usuário - Username. Produto - "BOLSAS", "BOLSAS_DIFERIDO" e "BOLSAS_SNAPSHOT" para ativos e "BROADCAST" para noticias e paginas fixas. Pagina - "S" para ativos e "W" para notícias e paginas fixas.	Retorna as permissões de um usuário	Comando: 1\tEXEC\tAEUSER PERM\tGetPermUser GetPermUser(Edu, BOLSAS_DIFERIDO, S) \n Resposta: C\t1\t6\tNYSE\t0

6.5 – Comando REQ

O comando REQ faz a requisição de tabelas e dados de ativos liberados para o usuário. O comando REQ permite a requisição de qualquer item fornecido pelo servidor, mas a título de exemplo vamos nos concentrar na requisição de tabelas e ativos, que são os mais utilizados.

Requisição de Tabelas

As tabelas mais utilizadas são as de bolsas e ativos. A primeira contém os dados das bolsas liberadas para o usuário, a segunda contém os ativos. A requisição de tabelas pode ser feita dentro do serviço BC|TABELA.

Exemplo 1: Requisição da tabela de bolsas

Comando: 1\tREQ\tBC|TABELA\tBOLSAS\n

O retorno da informação segue o formato:

```
C\tid\tbytes\t <NOME DA BOLSA>\t<CÓDIGO DA BOLSA>\n<NOME DA BOLSA>\t<CÓDIGO DA BOLSA>\n...
```

- <NOME DA BOLSA> - é o nome por extenso da bolsa
- <CÓDIGO DA BOLSA> - é o código que estará atrelado a um ativo.

Para o comando apresentado anteriormente a resposta seria:

```
C\t1\t228\tBMF\t002\tAE-TAXAS\t...
```

Exemplo 2: Requisição de tabelas de ativos

A tabela de ativos contém os dados dos ativos e os relacionam com as bolsas e HEADERS.

Comandos:

```
1\tREQ\tBC|TABELA\tTABELA\n
```

```
1\tREQ\tBC|TABELA\tTABELA_DIFERIDO\n
```

```
1\tREQ\tBC|TABELA\tTABELA_SNAPSHOT\n
```

O retorno da informação segue o formato:

```
C\tid\tbytes\t<CÓDIGO DA BOLSA>;<HEADER>;<CÓDIGO DO ATIVO>;<CGENÉRICO>\t...
```

- <CÓDIGO DA BOLSA> - é o código da bolsa a qual bolsa o ativo pertence,
- <HEADER> - pode ter os valores M para Mercadorias, A para Ações e T para Taxas

- <CÓDIGO DO ATIVO> - Código do ativo por exemplo PETR4
- <CÓDIG GENÉRICO> - O código genérico do ativo não é obrigatório, mas representa um alias para o ativo original.

Para o comando apresentado anteriormente a resposta seria:

```
C\t1\116818\t0002;M;DOLX09C2700\t...
```

Requisição de Ativos

A requisição de ativos solicita ao servidor os dados mais atuais de determinado ativo. A requisição de ativos pode ser feita dentro dos serviços BC|ATIVO (tempo real), BC|ATIVO_DIFERIDO e BC|ATIVO_SNAPSHOT.

Exemplo 1: Requisitando os campos do ativo

Os campos do ativo variam conforme o header ao qual ele esteja cadastrado (ACOES, MERCS ou TAXAS).

Comando: id\tREQ\tBC|ATIVO\tACOES\n

Resposta: C\t1\t222\tAtivo\tULT\tHOR\t...

Exemplo 2: Requisitando valores dos campos

É importante notar que os valores vêm na mesma posição de onde vieram os nomes dos campos.

Comando: id\tREQ\tBC|ATIVO\tPETR4\n

Resposta: C\t1\t547\tPETR4\t32,33\t16:40\t...

Utilizando o HEADER como gabarito para os valores

O HEADER deve ser utilizado como gabarito para os valores dos ativos a tabela abaixo ilustra como o HEADER indica o posicionamento dos valores. Essa tabela é exatamente o resultado dos comandos anteriormente citados.

Acoes

ATIVO	ULT	HOR	VAR	MAX	MIN	FEC	ABE	OCP	OVD	NEG	QUL	MED	VOC	VOV
PETR4	34,67	18:13	0,06	34,78	34,11	34,65	34,38	34,66	34,67	15908	900	34,47	5400	8800

Os campos de cada HEADER, bem como suas descrições estão no Apêndice 1.

6.6 – Comando START

O comando START solicita ao servidor que inicie o envio de mensagens (advices) a cada atualização do item que foi solicitado. Esse comando pode ser usado em diversas situações. Colocamos aqui alguns usos básicos desse comando. Abordamos a parte de notícias, ativos e alertas de alteração de tabelas. A tabela abaixo mostra os exemplos de uso do START.

Utilização do comando START

Estrutura do comando: **<ID>**\tSTART\t**<SERVICO>**\t**<PARAM1>**\t...**<PARAMn>**\n

Respostas:

Resposta de comando e de Advices

Serviço	Parâmetros	Descrição	Exemplo de aplicação
API API	MANCHETES	Últimas manchetes recebidas pelo servidor	Comando: 1\tSTART\tAPI API\tManchetes\n Resposta: C\t1\t3\tOK A\tServiço\tItem\tTamanho\tDados Exemplo: A\tAPI API\tManchetes\t111\t040509\tJN/Energy\t...\n
AEUSER AL TERA	ALTERA	Indica que houve uma mudança na tabela de ativos; adição, remoção, alteração, etc. Não deixe de ver observação ao final dessa sessão.	Comando: 1\tSTART\tAEUSER ALTERA\tALTERA\n Resposta: C\t1\t3\tOK\0 (sucesso) ou C\t1\t4\tNOK\0 (sem sucesso). A\tServiço\tItem\tTamanho\tDados Exemplo: A\tAEUSER ALTERA\tALTERA\t111\t ... \n
BC ATIVO, BC ATIVO_ DIFERIDO ou BC ATIVO_ SNAPSHOT	Código do Ativo, por exemplo PETR4	Inicia o envio de informações de um ativo.	Comando: 1\tSTART\tBC ATIVO\tPetr4\n Resposta: C\t1\t3\tOK A\tServiço\tItem\tTamanho\tDados Exemplo: A\tBC ATIVO\tPETR4\t305\tPETR4\t31.27\t...\n
	* o asterisco significa todos	Inicia o envio de informações de todos os ativos liberados para o usuário. O envio das informações é feito somente quando há atualização de algum valor do ativo. Para obter a última posição, deve se utilizar o REQ.	Comando: id\tSTART\tBC ATIVO\t*\n Resposta: C\tid\tbytes\tStatus As respostas de Advise serão enviadas a medida que os ativos tiverem seus valores atualizados, sendo uma para cada ativo.
	Lista de códigos de ativos.	Inicia o envio de informações dos ativos que estejam na lista.	Comando: id\tSTART\tBC ATIVO\tPETR4\tVALE5\n Resposta: C\tid\tbytes\tStatus As respostas de Advise serão enviadas a medida que os ativos tiverem seus valores atualizados, sendo uma para cada ativo.

Serviço	Parâmetros	Descrição	Exemplo de aplicação
BC LIVRO	Código do Ativo.OCP ou Código do Ativo.OVD	Inicia o envio de informações de livro de ofertas.	Comando: 1\tSTART\tBC ATIVO\tPetr4.OVD\n Resposta: C\t1\t3\tOK A\tServiço\tItem\tTamanho\tDados Exemplo: A\tBC LIVRO\tPETR4.OVD\t274\tA\t31.29\t...

Observação: Monitorar alteração de permissões AEUSER|ALTERA

Para monitorar alterações de permissões deve-se utilizar o comando START com o serviço AEUSER|ALTERA e parâmetro ALETRA.

Exemplo:

```
Id\tSTART\tAEUSER|ALTERA\tALTERA\n
```

A resposta será no formato advise sendo que o campo <DADOS> terá o seguinte formato:

<DADOS> = **Tipo\tUSER**

onde :

Tipo - Tipo de mudança no permissionamento, podendo ser :

"N" - Foi cadastrado ou descadastrado algum produto, serviço ou categoria (para este usuário ou para todos). Quando ocorrer este recebimento, o display deve ressolicitar todas as informações referentes ao tópico API (produtos, serviços e categorias).

"A" - Alguma informação referente aos ativos foi alterada. Neste caso o display deve ressolicitar a tabela de ativos, pois não está mais válida. De posse desta nova tabela, ele pode saber quais ativos não estão mais permissionados, ou os novos ativos disponíveis. Qualquer solicitação de algum ativo que tenha perdido permissionamento retornará N/A.

"R" - Idem para páginas em Real Time. Deve ser pedida a lista de páginas disponíveis novamente. Algumas páginas em advise podem não ser mais válidas.

"G" - Ocorreram mudanças em dois ou mais dos itens anteriores. Todas as informações contidas no display devem ser ressolicitadas.

USER - Usuário afetado. Quando as alterações afetam a todos os usuários (Cadastramento de um novo ativo), este campo vai preenchido com o valor "@everybody@".

6.7 – Comando STOP

Para cada item que o cliente faz o START, deve ser feito um STOP quando a informação não for mais desejada. A tabela abaixo mostra o uso do STOP.

Utilização do comando STOP

Estrutura do comando: **<ID>**\tSTOP\t<SERVICO>\t<PARAM1>\t...<PARAMn>\n

Respostas:

Resposta de comando

Serviço	Parâmetro	Descrição	Exemplo de aplicação
API API	MANCHETES	Interrompe o envio das manchetes.	Comando: 1\tSTOP\tAPI API\tManchetes\n Resposta: C\t1\t3\tOK
AEUSER ALTERA	ALTERA	Interrompe o envio de informações de alterações de tabelas de ativos	Comando: 1\tSTOP\tAEUSER ALTERA\tALTERA\n Resposta: C\t1\t3\tOK\0 (sucesso) ou C\t1\t4\tNOK\0 (sem sucesso).
BC ATIVO, BC ATIVO_DI FERIDO ou BC ATIVO_SN APSHOT	Código do ativo	Interrompe o envio de informações do ativo.	Comando: 1\tSTOP\tBC ATIVO\tPetr4\n Resposta: C\t1\t3\tOK
	* o asterisco significa todos	Interrompe o envio de informações de todos ativos.	Comando: 1\tSTOP\tBC ATIVO\t*\n Resposta: C\t1\t3\tOK
	Lista de ativos	Interrompe o envio dos ativos que estejam na lista.	Comando: 1\tSTOP\tBC ATIVO\tPETR4\tVALE5\n Resposta: C\t1\t3\tOK
BC LIVRO	Código do Ativo.OCP ou Código do Ativo.OVD	Interrompe o envio de informações de livro de oferta.	Comando: 1\tSTOP\tBC ATIVO\tPetr4.OVD\n Resposta: C\t1\t3\tOK\0 Comando: 1\tSTOP\tBC ATIVO\tPetr4.OCP\n Resposta: C\t1\t3\tOK\0

7 – Advise

7.1 - @SERVICE_ACTIVE@

Válido para todos os serviços (ex. AEUSER|PERM, BC|ATIVO, etc.)

Descrição:

O item "@SERVICE_ACTIVE@" pode ser assinado (comando **START**, para recebimento de *advices*) ou consultado (comando **REQ**) para qualquer serviço acessível via servidor **AEDataFeed**, e permite aos aplicativos clientes identificarem o estado de disponibilidade (ativo/inativo) de um serviço .

Formato: <flag_ativo>\t<módulo>\t<descrição> onde

- <flag_ativo>: **1** (serviço ativado) ou **0** (serviço desativado)
- <módulo>: nome do módulo que gerou o evento de ativação/desativação (*)
- <descrição>: descrição do motivo causador do evento de ativação/desativação (*)

(*): Campos opcionais, podem não existir dependendo da fonte geradora do evento

7.2 - DISCONNECTUSER

AEUSER|PERM

Descrição:

Advise para notificar aplicativos clientes sobre desconexão de usuários. O aplicativo cliente deve remover das suas estruturas de sessão o usuário desconectado e/ou alterar o estado do usuário para desconectado. Não é necessário chamar **CloseUserSession** em resposta a este advise.

Formato: <motivo>\t<usuário> onde

- <motivo>: qual tipo de evento causou a desconexão. Valores conhecidos hoje (novos valores poderão ser adicionados no futuro):
 - **LOGOFF**: sessão fechada normalmente
 - **TIMEOUT**: sessão fechada por timeout
 - **DESCONECTA**: sessão desconectada pelo administrador ou pela BCAA
 - **DESCADASTRA**: usuário foi descadastrado
- <usuário>: login do usuário cuja sessão foi desconectada

7.3 - PERMUSER

AEUSER|PERM

Descrição:

Advise para notificar aplicativos clientes sobre repermissionamento de usuários na BCAA. O aplicativo cliente, em momento oportuno, deve solicitar as permissões necessárias (por exemplo, invocando novamente o método `OpenUserSession` para usuários abertos através desse mecanismo).

Formato: **<usuário>** onde

- **<usuário>**: login do usuário cujas permissões foram alteradas

Obs:

- Novos campos, além de **<usuário>**, podem ser adicionados a este item no futuro com separador `\t` (TAB), portanto o parser do aplicativo cliente deve estar preparado para ignorar campos desconhecidos

Apêndice 1 – Campos de Ativos

Ativos são os papéis negociados em bolsa, cujos valores (cotações) são divulgados em Tempo Real pela BroadCast e mostrados de várias maneiras pelo AE Broadcast. Cada ativo tem um código, um Nome Completo e seus valores de cotações. Por exemplo, as ações da companhia Petrobras negociados na Bolsa de São Paulo são divulgadas como ativo com o código PETR4 e o nome completo PETROBRAS PN.

Os ativos são divididos em 3 categorias as quais chamamos de HEADERS: Ações, Mercadorias e Taxas. A tabela abaixo mostra os campos de cada HEADER bem como sua descrição.

Ações	Mercs	Taxas	Descrição
ABE	ABE	ABE	abertura
AN1	AN1	AN1	Times&Trades (4 últimos negócios)
AN2	AN2	AN2	Times&Trades (4 últimos negócios)
AN3	AN3	AN3	Times&Trades (4 últimos negócios)
AN4	AN4	AN4	Times&Trades (4 últimos negócios)
ATIVO	ATIVO	ATIVO	código do ativo
CCP	CCP		Código da corretora de compra
CVD	CVD		Código da corretora de venda
DCR	DCR	DCR	Dias corridos até o vencimento
DFC	DFC	DFC	Data do último fechamento
DFE			Data do fechamento do pregão eletrônico
DRF	DRF	DRF	Data de referência
DUT	DUT	DUT	Dias úteis até o vencimento
DVC	DVC		Data de vencimento
EST	EST		Estado
FCE			Fechamento do pregão eletrônico
FEC	FEC	FEC	fechamento
HO1	HO1	HO1	Times&Trades (Horário dos último negócios)
HO2	HO2	HO2	Times&Trades (Horário dos último negócios)
HO3	HO3	HO3	Times&Trades (Horário dos último negócios)
HO4	HO4	HO4	Times&Trades (Horário dos último negócios)
HOR	HOR	HOR	hora da última cotação
INC			Indexador de correção
MAX	MAX	MAX	máxima
MED	MED		média
MIN	MIN	MIN	mínima
NCC	NCC		Código da corretora compradora
NCV	NCV		Código da corretora vendedora
NEG	NEG		número de negócios
NOM	NOM	NOM	Nome do ativo
NSQ	NSQ		Número de saques
OCP	OCP	OCP	oferta de compra
OVD	OVD	OVD	oferta de venda
PEX	PEX	PEX	Preço de exercício
PPI			Participação percentual no IBOV

PRR	PRR		Preço de referência
PTR			Preço teórico
QCC	QCC		Quantidade da oferta da corretora de compra
QCV	QCV		Quantidade da oferta da corretora de venda
QTR			Quantidade teórica
QTT			quantidade total
QUL			quantidade do último negócio
TAV	TAV	TAV	tipo de ativo
TLN			Tamanho do lote de negociação
TND	TND	TND	tendência
TNG	TNG	TNG	Tipo do negócio
ULT	ULT	ULT	preço da última cotação
VAR	VAR	VAR	variação
VO1	VO1	VO1	Times&Trades (Volume dos últimos negócios)
VO2	VO2	VO2	Times&Trades (Volume dos últimos negócios)
VO3	VO3	VO3	Times&Trades (Volume dos últimos negócios)
VO4	VO4	VO4	Times&Trades (Volume dos últimos negócios)
VOC	VOC		volume da oferta de compra
VOR	VOR		Volume em Reais
VOV	VOV		volume da oferta de venda
	AJA		Ajuste anterior
	AJU		ajuste
	CAB		contratos em aberto
	CNG		contratos negociados
	ENI		Estado de negociação do instrumento
	HOC	HOC	horário da oferta de compra
	HOV	HOV	horário da oferta de venda
	LMA		Limite Móvel Máximo
	LMI		Limite Móvel Mínimo
	OSA		Oscilação máxima
	OSI		Oscilação mínima
	RLM		Referência do Limite Móvel
	YLD	YLD	Yield