



Complemento
liberdade e tecnologia

Access Control Lists - ACL

Treinamento OTRS Help Desk



Sumário

Introdução.....	3
Seções compõe as ACLs.....	4
Properties.....	4
PropertiesDatabase.....	4
Possible.....	4
PossibleNot.....	4
Propriedades.....	6
Action.....	6
Expressões Regulares.....	6
Parâmetros possíveis.....	7



Introdução

As ACL's servem para incrementar o sistema de permissões do OTRS. Com elas é possível restringir escolhas de atributos do ticket ou ações possíveis de serem tomadas de acordo com as propriedades atuais do mesmo (fila atual, estado etc).

Atualmente criamos as ACLs com implementação de códigos no arquivo Kernel/Config.pm (recomendado) e não há interface gráfica para isso. Com sua utilização, é possível inclusive implementar pequenos workflows no sistema.

Nas primeiras versões do OTRS, as ACLs eram implementadas usualmente no arquivo de configuração do Sistema, o Config.pm.

As versões mais recentes permitem a criação de ACLs a partir da interface, a sintaxe porém, se mantém muito parecida

Vejamos um exemplo de ACL escrita em arquivo de configuração que restringe um chamado de prioridade alta (5) para que seja permitido movê-lo apenas para uma fila chamada Alerta:

```
# ticket acl
$self->{TicketAcl}->{'ACL-Nome-2'} = {
  # match properties
  Properties => {
    # current ticket match properties
    Ticket => {
      Queue => ['Raw'],
      Priority => ['5 very high'],
    }
  },
  # return possible options (white list)
  Possible => {
    # possible ticket options (white list)
    Ticket => {
      Queue => ['Alerta'],
    },
  },
};
```

Neste exemplo, é bom que se deixe claro que a única ação que restringimos foi a alteração de fila. As outras ações continuam possíveis de acordo com as parametrizações e permissões do usuário. Por exemplo, ainda é possível adicionar notas ao chamado, respondê-lo, mudar seu status para qualquer um disponível no sistema.



É possível notar que existem dois blocos de código neste exemplo acima. O primeiro com comentário “# match properties” é a definição das propriedades atuais do ticket, como se fosse um filtro onde definimos em que ocasiões essa ACL será aplicada.

No segundo bloco definimos as restrições ou permissões que os tickets que “cairem” nesta ACL sofrerão.

Para criar ACLs através da interface do OTRS, acesse **Administração → Listas de Controle de Acesso**

Adicione uma nova ACL clicando em “Criar nova ACL”.

Preencha os campos a seguir:

★ Nome:

Comentário:

Descrição:

Parar Após Encontrar:

★ Validade:

ou

Dicas:

- As ACLs são interpretadas em ordem alfabética, por tanto, uma ACL que comece com a letra “Z” pode invalidar as condições de uma ACL que começa com a letra “A”.
- Altere a validade da ACL para “Válido”

Após preencher esta tela, clique em Salvar.

Você terá então que começar a compor sua ACL na tela abaixo:



Configurações de casamento

- ▼ Properties
 - ⊕ -
- ▼ PropertiesDatabase
 - ⊕ -
 -

Configurações de alteração

- ▼ Possible
 - ⊕ -
- ▼ PossibleAdd
 - ⊕ -
- ▼ PossibleNot
 - ⊕ -

Um exemplo:

▼ Editar estrutura da ACL

Configurações de casamento

- ▼ Properties
 - ▼ Ticket
 - Queue:
 - Raw x Exact match ▾
 - Priority:
 - 5 very high x Exact match ▾
 - ⊕
 - ⊕ -
 -

Configurações de alteração

- ▼ Possible
 - ▼ Ticket
 - Queue:
 - Alert x Exact match ▾
 - ⊕
 - ⊕ -
 -

Vamos entender a seguir, as seções que compõe uma ACL.

Seções compõe as ACLs



Properties

Aqui definimos os critérios, ou seja, o filtro que definirá se esta ACL será executada ou não. No exemplo abaixo, definimos que os chamados serão afetados caso o atendente selecione a prioridade '5 muito alta' na tela em que está.

```
Properties => {  
  Ticket => {  
    Priority => ['5 very high'],  
  }  
},
```

PropertiesDatabase

Assim como “Properties”, esta seção é um filtro. A diferença entre ambas é que Properties realiza a comparação com os valores que estão na tela, durante uma modificação por exemplo de valores, reclassificação de chamados etc, enquanto PropertiesDatabase realiza uma comparação com as informações que estão no banco de dados do chamado. No exemplo abaixo, os chamados que possuem prioridade muito alta serão afetados, independentemente das alterações em curso na tela.

```
PropertiesDatabase => {  
  Ticket => {  
    Priority => ['5 very high'],  
  }  
},
```

Possible

Define quais são os valores que poderão ser escolhidos. Outros valores serão excluídos da exibição. No exemplo abaixo, os chamados que corresponderem aos critérios definidos em Properties ou PropertiesDatabase, poderão apenas serem movidos para a fila 'Alerta' e para nenhuma outra. Lembrando que aqui estamos restringindo apenas a fila para onde ele poderá ser movido, todas as outras funções tais como criar nota, responder, bloquear, não são afetadas.

```
Possible => {  
  Ticket => {  
    Queue => ['Alerta'],  
  },  
},
```

PossibleNot

Define os valores que não poderão ser exibidos. Todos os outros valores serão. No exemplo abaixo, os chamados poderão ser movidos para quaisquer filas, menos para a fila Alerta.

```
PossibleNot => {  
  Ticket => {  
    Queue => ['Alerta'],  
  },  
},
```



},

Vamos simular então uma ACL onde o atendente nunca poderá fechar um chamado se o mesmo estiver na fila Raw com prioridade 5. Ficaria assim:

```
# ticket acl
$Self->{TicketAcl}->{'ACL-Alerta5'} = {
  # match properties
  Properties => {
    # current ticket match properties
    Ticket => {
      Queue => ['Raw'],
      Priority => ['5 very high'],
    }
  },
  # return possible options (white list)
  Possible => {
    # possible ticket options (white list)
    Ticket => {
      Queue => ['Raw'],
    },
    Action => {
      AgentTicketClose => 0,
    },
  },
  PossibleNot => {
    # possible not ticket options
    Ticket => {
      State => ['closed successful', 'closed unsuccessful'],
    },
  },
},
};
```

PossibleAdd

Adiciona valores possíveis a um atributo, sem ser restritivo. Um exemplo: permitir que um usuário encerre o chamado apenas após este ser classificado:

1. Crie uma ACL que retire os estados “[RegExp]^closed” usando o PossibleNot.
2. Crie uma ACL que adiciona os mesmos estados usando o PossibleAdd, quando o serviço for definido.



Propriedades

Action

No exemplo acima, vimos o elemento “Action”. Ele define se uma tela do sistema ficará disponível ou não para o atendente. Porém, ela deve ser declarada apenas na seção “Possible”, mesmo que o objetivo seja negar a exibição do módulo. Para habilitar, coloque o valor 1, para negar, coloque o valor 0, como no exemplo acima:

```
Action => {  
  AgentTicketClose => 0,  
},
```

Expressões Regulares

Também é possível utilizar expressões regulares. No exemplo abaixo (retirado da documentação oficial), exibimos apenas serviços que comecem com a palavra “Hardware”, para um ticket estiver na fila HW ou uma de suas subfilas:

```
$Self->{TicketAcl}->{'100-Only-Hardware-Services-for-HW-Queues'} = {  
  # match properties  
  # note we don't have "Ticket => {" because there's no ticket yet  
  Properties => {  
    Queue => {  
      Name => ['[RegExp]HW'],  
    }  
  },  
  # return possible options  
  Possible => {  
    # possible ticket options  
    Ticket => {  
      Service => ['[RegExp]^(Hardware)'],  
    },  
  },  
};
```

Vale a pena lembrar que os serviços começados com a palavra “Hardware” continuaram sendo exibidos em outras filas. Foi utilizando esse tipo de ACL que construí um módulo que permite escolher os serviços que queremos exibir em cada uma das filas do sistema.



Parâmetros possíveis

Veja no link abaixo a referência de todos os parâmetros possíveis para as ACLs

<http://otrs.github.io/doc/manual/admin/5.0/en/html/customization.html#acl>