





Nota

Antes de utilizar estas informações e o produto suportado por elas, leia as informações em “Avisos”, na página 125.

Sobre Estas Informações

Estas informações apresentam uma visão geral da Tecnologia ACT da IBM e sua função no processamento de eventos complexos. A linguagem de regra da Tecnologia ACT baseia-se em XML para criar regras de correlação de eventos. Estas informações destinam-se a autores de regras que precisam compreender como criar regras para correlacionar eventos para suas organizações de negócios.

Índice

Sobre Estas Informações	iii
--	------------

Parte 1. Guia do Autor de Regras . . .	1
---	----------

Capítulo 1. Introdução.	3
--	----------

Capítulo 2. Visão Geral da Linguagem de Regra	5
--	----------

Anatomia de uma Regra	5
Ciclo de Vida de uma Regra	7
Organização de Regras	9
Padrões de Regras	11
Padrão de Coleta	11
Padrão de Computação	12
Padrão de Duplicata	12
Padrão de Filtragem	13
Padrão de Sequência	14
Padrão de Limite	17
Padrão de Cronômetro	19
Aspectos Comuns e Exclusivos de Padrões de Regras Diferentes	19
Expressões.	20
Importando e Acessando Módulos e Objetos Externos	21
Inicializando e Acessando Variáveis	22
Acessando Informações Relacionadas a Eventos Boas Práticas para Codificação de Expressões	23
Variáveis	24
Tipos de Dados das Variáveis da Tecnologia ACT	26
Contextos de Expressões nos quais as Variáveis são Válidas	26
Variável act_event	27
Variável act_eventCount	28
Variável act_eventList	29
Variável act_lib	29
Variável act_location	31
Variável act_nodeName	32
Variável act_threshold	33
Fluxo de Eventos por meio de um Conjunto de Regras	33

Capítulo 3. Visão Geral da Gravação de Regras.	35
---	-----------

Planejamento para a Correlação de Eventos.	35
Projetando as Regras para Correlacionar os Eventos	37
Introdução ao Construtor de Regras	38
Configurando a Perspectiva no Eclipse Workbench	38
Configurando as Preferências	39
Criando um Projeto para Armazenar um Arquivo de Conjunto de Regras	39
Criando um Conjunto de Regras	40
Criando um Bloco de Regra	40
Criando uma Regra.	41

Validando um Conjunto de Regras	41
Compilando um Conjunto de Regras	42
Atualizando um Conjunto de Regras	42
Incluindo Fragmentos em Expressões Dentro das Regras	43

Parte 2. Referência do Autor de Regras	45
---	-----------

Capítulo 4. Resumo da Organização do Conjunto de Regras	47
--	-----------

Resumo do Conjunto de Regras	47
Resumo do Bloco de Regra	47
Resumo da Regra de Coleta	48
Resumo da Regra de Computação.	50
Resumo da Regra de Duplicata.	51
Resumo da Regra de Filtragem.	52
Resumo da Regra de Sequência.	53
Resumo da Regra de Limite	55
Resumo da Regra de Cronômetro	56

Capítulo 5. Referência de Elemento de Linguagem	59
--	-----------

Elemento action	60
Elemento activateOnEvent	61
Elemento activationByGroupingKey	62
Elemento activationInterval	69
Elemento activationTime	71
Elemento after	72
Elemento attributeAlias	73
Elemento attributeName	74
Elemento booleanThreshold	74
Elemento collectionRule	75
Elemento comment	76
Elemento computationRule	77
Elemento computedThreshold	78
Elemento computedValue.	80
Elemento computeFunction	81
Elemento dateTime	82
Elemento deactivateOnEvent	83
Elemento duplicateRule	84
Elemento eventAttribute	85
Elemento eventCountThreshold.	86
Elemento eventSelector	88
Elemento eventType	90
Elemento filteringPredicate	91
Elemento filterRule	91
Elemento groupingKey	93
Elemento import.	95
Elemento inactiveWhenLoaded	96
Elemento lifeCycleActions	96
Elemento never	97
Elemento onActivation	97
Elemento onDeactivation	98

Elemento onDetection	99
Elemento onLoad	99
Elemento onNextEvent	100
Elemento onTimeOut.	101
Elemento onTimeWindowComplete	102
Elemento onUnload	102
Elemento ruleBlock	103
Elemento ruleSet	104
Elemento runUntilDeactivated.	105
Elemento sequenceRule	106
Elemento start	108
Elemento stop	109
Elemento stopAfter	109

Elemento thresholdRule	111
Elemento timeInterval	112
Elemento timerRule	113
Elemento timeWindow	115
Elemento variable	115
Elemento varInitializer	118
Elemento whenLoaded	118

Capítulo 6. Glossário 121

Apêndice. Avisos. 125

Marcas Registradas	126
------------------------------	-----

Parte 1. Guia do Autor de Regras

Capítulo 1. Introdução

Esta introdução descreve brevemente o processamento de eventos complexos (também conhecidos como CEP) e fornece uma visão geral da Tecnologia ACT e de sua função no processamento de eventos complexos.

O Ambiente de Negócios Atual

Hoje, as organizações comerciais e governamentais dependem do processamento de informações eletrônicas por meio de redes de computadores e especialmente por meio da Internet. Com tecnologias adicionais, como computação em grade, as organizações executam aplicativos críticos a qualquer momento e colocam no mundo. Os processos de negócios, a atividade, a infra-estrutura e, portanto, nossa sociedade global, dependem da camada de TI (Tecnologia de Informações) das organizações.

As organizações precisam saber o que está acontecendo com seus negócios em tempo integral. Por exemplo, elas precisam saber se os aplicativos críticos estão disponíveis e funcionando adequadamente e como detectar e impedir uma provável crise nos processos de negócios, na atividade ou na infra-estrutura. Se ocorrer uma crise, elas precisarão entender o problema imediatamente, como corrigi-lo e qual foi a causa.

A importância da maioria dos eventos relacionados aos processos de negócios, à atividade e à infra-estrutura nunca é reconhecida ou entendida, porque a quantidade de informações é muito grande e muito difícil assimilar, uma vez que é apresentada em partes individuais não relacionadas. No entanto, se os eventos forem agregados e correlacionados para que seus relacionamentos possam ser facilmente entendidos, eles poderão produzir uma infinidade de informações.

O objetivo do processamento de eventos complexos é obter informações mais adequadas sobre os eventos em tempo real.

Processamento de Eventos Complexos

Um *evento* é simplesmente uma notificação sobre algo ocorrido.

Processamento de eventos complexos é a derivação de eventos de alto nível da análise, da correlação e do resumo de eventos de baixo nível em sistemas orientados a eventos. Esses eventos de alto nível, chamados de eventos complexos, são adequados para notificar pessoas sobre as oportunidades ou os problemas de negócios em termos de fácil entendimento ou para acionar os processos automatizados. As organizações podem, então, operar com maior eficiência, com aviso antecipado de oportunidades ou de problemas potenciais e com melhor entendimento das causas raízes que alteram as condições em seus processos de negócios, atividade ou infra-estrutura.

Correlação de eventos é o processo de definição e detecção de padrões em fluxos de eventos em tempo real e de ações de implementação em resposta a eventos relacionados. É utilizada para identificar um problema com base em seus sintomas detectados. Os eventos podem ser correlacionados por causa, por tempo, por associação ou por combinações deles. A correlação de eventos é uma parte integral do processamento de eventos complexos.

A Tecnologia ACT

A Tecnologia ACT utiliza regras para detectar padrões em fluxos de eventos em tempo real. Essa tecnologia baseia-se no entendimento de que, em muitos casos, as ações de respostas não devem ser acionadas por um único evento de baixo nível, mas por uma composição complexa de eventos que estão acontecendo em horas e em contextos diferentes. A Tecnologia ACT utiliza os relacionamentos entre os eventos para fornecer reconhecimento das oportunidades e dos problemas de negócios. Por exemplo, com base no reconhecimento de negócios obtido por meio da correlação de eventos em tempo real, uma organização pode executar os seguintes tipos de ações:

- Oferecer remessa com desconto para alguns ou para todos os clientes durante uma venda em feriado.
- Nos próximos 30 dias, calcular o custo da remessa com base na transportadora, no valor do pedido e na quantidade do pedido.
- Enviar aos clientes que comprarem mercadorias com valor superior a USD500, entre 1 de julho de 2005 e 31 de dezembro de 2005, um certificado de presente de USD25.
- Notificar um administrador se algum processamento de pedido não for concluído dentro de 36 horas.
- Notificar um administrador se forem detectadas mais de quatro tentativas de login no mesmo computador no intervalo de 30 segundos.

A Tecnologia ACT consiste nos seguintes itens primários:

linguagem de regras da Tecnologia ACT

Uma linguagem baseada em XML para gravar regras para correlacionar eventos. Essas regras poderão então ser implementadas em ambientes de tempo de execução da Tecnologia ACT.

mecanismo da Tecnologia ACT

O componente Tecnologia ACT que processa eventos de acordo com a saída do compilador da Tecnologia ACT.

construtor de regras da Tecnologia ACT

Uma GUI para gravar regras de correlação na linguagem de regras da Tecnologia ACT.

Um ambiente de tempo de execução da Tecnologia ACT é um aplicativo no qual o mecanismo da Tecnologia ACT está incorporado.

Capítulo 2. Visão Geral da Linguagem de Regra

Esta visão geral descreve os conceitos-chave da linguagem de regra da Tecnologia ACT.

Um padrão de regra é a representação de uma situação de correlação de eventos (como uma condição de limite ou uma detecção de eventos de duplicata). A linguagem das regras da Tecnologia ACT inclui sete padrões de regras testadas para representar a maioria das situações de correlação de eventos que os clientes da IBM precisam resolver. Seis deles definem as regras com preservação de estado e o outro define uma regra sem preservação de estado.

As regras com preservação de estado correlacionam vários eventos que ocorrem durante um período de tempo específico e geram uma resposta para eles. As regras sem preservação de estado processam apenas um único evento que atende a uma determinada condição e geram uma resposta para ele.

regra com preservação de estado

Uma regra que retém informações de estado, que são informações sobre as características de uma instância de regra, com o fim de agir em uma coleta de eventos em um período de tempo. As regras definidas por um dos padrões de regras a seguir são regras com preservação de estado: coleta, computação, duplicata, seqüência, limite ou cronômetro.

regra sem preservação de estado

Uma regra que não retém informações de estado e, portanto, podem agir apenas em um evento de cada vez. Uma regra definida pelo padrão de filtragem é uma regra sem preservação de estado.

Referência Relacionada

Capítulo 4, “Resumo da Organização do Conjunto de Regras”, na página 47
Esta referência lista todos os elementos da linguagem de um conjunto de regras, um bloco de regra e cada tipo de regra. Ela serve como referência rápida para codificação de um conjunto de regras.

Capítulo 5, “Referência de Elemento de Linguagem”, na página 59
Essa referência descreve os detalhes dos elementos de linguagem no esquema XML da linguagem de regras da Tecnologia ACT. Os elementos de linguagem são listados em ordem alfabética e os atributos disponíveis para cada elemento são descritos no tópico desse elemento.

Capítulo 6, “Glossário”, na página 121
Este glossário contém os termos e as definições de conceitos importantes na Tecnologia ACT.

Anatomia de uma Regra

As partes mais básicas de uma regra são a seleção de evento, a chave de agrupamento, o espaço de tempo para regras com preservação de estado, a resposta da regra, o intervalo de ativação e as ações do ciclo de vida. Uma regra também inclui expressões e variáveis. Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra.

Seleção de evento

Os critérios de seleção de evento determinam quais eventos são aceitos para processamento pela regra. O elemento `<eventSelector>` define os critérios de seleção de evento para uma regra. A seleção de evento aplica-se a todas as regras, exceto àquelas definidas pelo padrão de cronômetro. Como a regra do cronômetro não processa eventos, ela não contém critérios de seleção de eventos.

Chave de Agrupamento

Normalmente, cada regra ativa possui uma instância de regra, ou cópia, em execução no mecanismo da Tecnologia ACT. No entanto, às vezes, a mesma regra é necessária para grupos diferentes de eventos, relacionados, com frequência, a diferentes grupos de recursos. A chave de agrupamento é um método para direcionar uma regra para criar uma instância de regra separada (ou uma cópia de si mesma) para cada grupo de eventos que compartilham características comuns.

A chave de agrupamento serve como forma adicional de seleção de evento. Se uma regra for definida com uma chave de agrupamento e receber um evento com a característica definida pela chave de agrupamento, o evento será enviado para a instância de regra que estiver processando os eventos que compartilham essa característica. Por exemplo, é possível definir uma regra que colete todos os eventos de segurança do tipo Falha na Auditoria e definir a chave de agrupamento para ser o atributo de nome do host de um evento. Agora, a regra pode ser utilizada várias vezes com uma cópia separada da regra executada para cada valor exclusivo do atributo de nome do host. Também é possível monitorar todos os sistemas que recebem o evento Falha na Auditoria para determinar se mais de 10 desses eventos ocorrem em um período de tempo de 2 minutos para cada nome do host.

O elemento `<groupingKey>` define a chave de agrupamento de uma regra e é válido para regras definidas pelos padrões de coleta, de computação, de duplicata, de sequência e de limite.

Espaço de tempo para regras com preservação de estado

Como as regras com preservação de estado correlacionam vários eventos que ocorrem durante um período de tempo específico, uma parte básica de uma regra com preservação de estado é o espaço de tempo, definido pelo elemento `<timeWindow>`. O espaço de tempo especifica o período de tempo durante o qual a regra com preservação de estado está sendo processada para corresponder a seu padrão.

Resposta da regra

As ações da resposta da regra definem as ações a serem executadas quando a regra conclui seu processamento. Cada um dos seguintes elementos de linguagem define um tipo diferente de ação da resposta da regra:

- `<action>` em `<onDetection>`
- `<action>` em `<onNextEvent>`
- `<action>` em `<onTimeOut>`
- `<action>` em `<onTimeWindowComplete>`

Os tipos de ações da resposta da regra disponíveis para uma regra dependem do padrão da regra.

Intervalo de ativação

O intervalo de ativação define quando uma regra está ativa e inativa. O elemento `<activationInterval>` define o intervalo de ativação de uma regra.

Uma regra pode ser ativada ou desativada em um momento exato distinto ou por um evento específico.

Se você especificar que uma regra deve ser ativada ou desativada em um momento exato distinto e por um evento específico, a regra será ativada ou desativada, o que ocorrer primeiro, o momento exato ou o recebimento do evento. No entanto, nesse caso, a regra poderá ser ativada ou desativada por vários eventos durante seu ciclo de vida. Por exemplo, uma regra pode ser ativada por um evento, desativada, ativada em um momento exato definido, desativada novamente e ativada por outro evento.

O elemento `<activationByGroupingKey>` é um que está contido no elemento `<activationInterval>`. O elemento `<activationByGroupingKey>` contém elementos que especificam os eventos que podem ativar e desativar uma instância de regra definida pelo elemento `<groupingKey>`.

Ações do ciclo de vida

As ações do ciclo de vida definem as ações a serem executadas nos quatro estágios primários a seguir no ciclo de vida de uma regra: carregamento, ativação, desativação e descarregamento.

O elemento `<lifeCycleActions>` contém os seguintes elementos que definem essas ações:

- `<action>` em `<onLoad>`
- `<action>` em `<onActivation>`
- `<action>` em `<onDeactivation>`
- `<action>` em `<onUnload>`

Ciclo de Vida de uma Regra

Cada estágio no ciclo de vida de uma regra pode ter várias causas e efeitos. Gravando e incluindo expressões nas ações do ciclo de vida (conforme definido pelo elemento `<lifeCycleActions>`), um autor de regras pode definir as ações a serem executadas em cada estágio.

Estágios no Ciclo de Vida de uma Regra

A seguir estão os quatro estágios primários no ciclo de vida de uma regra:

Carregamento

O carregamento da regra no mecanismo da Tecnologia ACT, que aciona as ações no elemento `<onLoad>`.

Ativação

A ativação da regra, que aciona as ações no elemento `<onActivation>`.

Desativação

A desativação da regra, que aciona as ações no elemento `<onDeactivation>`.

Descarregamento

O descarregamento da regra no mecanismo da Tecnologia ACT em execução, que aciona as ações no elemento `<onUnload>`.

Os estágios de ativação e desativação podem ocorrer várias vezes no ciclo de vida de uma regra, mas os estágios de carregamento e descarregamento ocorrem apenas uma vez.

Normalmente, não é necessário definir ações de ciclo de vida. A seguir, exemplos de quando você pode definir uma determinada ação de ciclo de vida:

- Quando uma determinada regra for carregada, talvez você queira criar uma conexão com um sistema externo (como um gerenciador de banco de dados) que precise ser acessado nessa regra. Quando essa mesma regra for descarregada, você poderá eliminar a conexão e executar os processos de limpeza necessários.
- Quando uma determinada regra for ativada, talvez você queira verificar se determinados recursos estão disponíveis para ela.
- Quando uma regra de limite for desativada, mas o limite não tiver sido atendido e o período de tempo ainda não tiver passado, talvez você queira redirecionar uma mensagem para alguém com essas informações.

Como a ativação e a desativação de uma regra podem ocorrer várias vezes no ciclo de vida, as ações codificadas para esses estágios poderão ser executadas com frequência.

As Causas e os Efeitos de cada Estágio de Ciclo de Vida

Tabela 1 lista as causas e os efeitos de cada estágio de ciclo de vida.

Tabela 1. Causas e Efeitos de cada Estágio de Ciclo de Vida

Estágio de ciclo de vida	Causas	Efeitos
Carregamento	Nas seguintes circunstâncias: <ul style="list-style-type: none">• Uma regra ou um bloco de regra é incluído ou substituído, causando o carregamento da nova regra ou regras.• O conjunto de regras é substituído no mecanismo da Tecnologia ACT, causando o carregamento das regras no novo conjunto de regras.	As ações no elemento <code><onLoad></code> são executadas.
Ativação	A regra é ativada. Uma regra pode ser ativada das seguintes maneiras: <ul style="list-style-type: none">• De acordo com as definições no elemento <code><activationInterval></code>• Por meio do método <code>activate()</code> disponível por meio da variável <code>act_lib</code>• Por meio das chamadas de aplicativos para o método <code>activate()</code> no mecanismo da Tecnologia ACT	Se a regra estiver inativa, as ações no elemento <code><onActivation></code> serão executadas.

Tabela 1. Causas e Efeitos de cada Estágio de Ciclo de Vida (continuação)

Estágio de ciclo de vida	Causas	Efeitos
Desativação	<p>A regra é desativada. Uma regra pode ser desativada das seguintes maneiras:</p> <ul style="list-style-type: none"> • De acordo com as definições no elemento <activationInterval>, exceto que o elemento <deactivateOnEvent> no elemento <activationByGroupingKey> não causa desativação da regra • Por meio do método deactivate() disponível por meio da variável act_lib • Por meio das chamadas de aplicativos para o método deactivate() no mecanismo da Tecnologia ACT 	<p>Se a regra estiver ativa, as ações no elemento <onDeactivation> serão executadas.</p>
Descarregamento	<p>Nas seguintes circunstâncias:</p> <ul style="list-style-type: none"> • O mecanismo da Tecnologia ACT é encerrado, causando o descarregamento das regras. • Uma regra ou um bloco de regra é removido ou substituído, causando o descarregamento da regra ou regras antigas. • O conjunto de regras é removido ou substituído no mecanismo da Tecnologia ACT, causando o descarregamento das regras no conjunto de regras antigo. 	<p>Se a regra estiver ativa, as ações no elemento <onDeactivation> serão executadas, seguidas pelas ações do elemento <onUnload>. Caso contrário, as ações no elemento <onUnload> serão executadas.</p>

Organização de Regras

A linguagem de regras da Tecnologia ACT organiza as regras em blocos que compõem um conjunto de regras.

Conjunto de regras

O conjunto de regras contém as regras, organizadas em blocos de regras, a serem executadas por um mecanismo da Tecnologia ACT. É a unidade de execução da regra. Cada mecanismo da Tecnologia ACT age em apenas um conjunto de regras de cada vez.

As regras contidas em um conjunto de regras são acionadas por eventos que são enviados para o mecanismo da Tecnologia ACT. Os eventos são entregues em seqüência para as regras apropriadas com base nos critérios de seleção de eventos de cada regra e é executada uma regra de cada vez. O mesmo evento pode ser aplicado e, portanto, acionar, várias regras. Essas regras não estão necessariamente relacionadas, mas poderiam estar.

A ordem dos blocos de regras e das regras em um conjunto de regras determina como os eventos fluirão pelo conjunto de regras.

As variáveis e as importações poderão ser definidas no nível do conjunto de regras para uso em expressões (código que contém lógica personalizada) no escopo do conjunto de regras. Uma importação é uma maneira específica da linguagem de programação acessar código externo. Um autor de regras pode definir a

importação de módulos externos (por exemplo, classes Java) para serem utilizados nas expressões dentro das regras.

Bloco de regra

O bloco de regra é a organizational unit para agrupamento de regras por função em domínios no conjunto de regras. Um domínio é a categoria à qual um grupo de regras é aplicado com base em sua função. Por exemplo, um domínio pode representar uma área geográfica específica, a disciplina de gerenciamento de TI (como detecção de segurança ou correlação de eventos da rede) ou a organização de negócios (como uma empresa específica ou uma divisão com uma empresa).

Os blocos de regras podem conter regras e outros blocos de regras. Como os blocos de regras podem ser aninhados, é possível construir uma hierarquia de regras. Por exemplo, um conjunto de regras pode conter um bloco de regra para correlação de eventos da rede e o bloco de regra para correlação de eventos da rede pode conter dois outros blocos de regras: um para a correlação da camada 2 e outro para a correlação de IP.

Portanto, um conjunto de regras fornece recursos de correlação de eventos para uma variedade de domínios e um bloco de regra fornece a organização para esses domínios diferentes que podem precisar de acesso a um conjunto semelhante de eventos.

As variáveis e as importações podem ser definidas no nível do bloco de regra para uso em expressões no escopo do bloco de regra. O escopo de um bloco de regra inclui quaisquer regras e outros blocos de regras contidos no bloco de regra.

Regra

A regra é a unidade de correlação utilizada para reconhecer relacionamentos entre os eventos e para executar as respostas apropriadas da regra. Uma regra é uma implementação de um dos sete padrões de regras a seguir e é organizada, de acordo com sua função, em um bloco de regra que faz parte de um conjunto de regras:

- padrão de coleta
- padrão de computação
- padrão de duplicata
- padrão de filtragem
- padrão de sequência
- padrão de limite
- padrão de cronômetro

Cada regra pode fornecer recursos exclusivos de correlação de eventos de acordo com seu padrão e as regras podem ser encadeadas por meio do redirecionamento de eventos. Esse encadeamento de regras permite combinar ou aninhar os recursos de correlação de eventos de diferentes padrões.

As variáveis podem ser definidas no nível de regra para uso em expressões no escopo da regra.

Resumo

Resumindo, o conjunto de regras é a unidade de execução, o bloco de regra é a organizacional unit e a regra é a unidade de correlação. Um conjunto de regras contém um ou mais blocos de regras, cada uma das quais pode conter blocos de regras adicionais. Cada um dos blocos de regras contém regras para um domínio específico. Os blocos de regras podem ser aninhados para construir uma hierarquia de regras. A ordem dos blocos de regras e das regras em um conjunto de regras determina como os eventos fluirão pelo conjunto de regras.

As variáveis e as importações podem ser definidas no nível do conjunto de regras ou do bloco de regra para uso em expressões nas regras. O escopo da variável ou da importação é o conjunto de regras ou o bloco de regra respectivo. As variáveis também podem ser definidas no nível da regra, mas isso limita seu escopo apenas à regra.

Padrões de Regras

Um padrão de regra é a representação de uma situação de correlação de eventos (como uma condição de limite ou uma detecção de eventos de duplicata). A linguagem de regra da Tecnologia ACT define os seguintes padrões de regras: coleta, computação, duplicata, filtragem, seqüência, limite e cronômetro.

O padrão de uma regra é correspondido quando a situação definida pela regra ocorre. Quando o padrão é correspondido, a regra conclui seu processamento executando as ações apropriadas de resposta da regra. Enquanto uma regra está ativa, o padrão de regra pode ser correspondido várias vezes.

As regras definidas pelo padrão de filtragem são as únicas regras sem preservação de estado na linguagem de regra. As outras regras são com preservação de estado.

Padrão de Coleta

A regra de coleta é definida pelo padrão de coleta. Ela coleta um grupo de eventos selecionados em um intervalo de tempo. É uma regra com preservação de estado.

Visão Geral

O padrão de coleta é utilizado para reunir eventos semelhantes em um período de tempo. O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>` na linguagem da regra.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de coleta, a resposta da regra é executada quando o espaço de tempo é concluído, conforme definido pelo elemento `<onTimeWindowComplete>`.

Exemplo de Uso desse Padrão de Regra

Um exemplo de uso do padrão de coleta é uma regra que faz o seguinte:

Ela coleta eventos que atendem aos critérios de um determinado seletor de eventos durante o período de tempo. Quando o período de tempo termina, ela resume os eventos coletados em um único evento que contém a contagem total de eventos e de informações de características sobre os eventos resumidos.

Referências relacionadas

“Resumo da Regra de Coleta” na página 48

Este resumo lista todos os elementos da linguagem da regra de coleta.

Padrão de Computação

Uma regra de computação é definida pelo padrão de computação. Ela aplica um cálculo (por meio de uma expressão) nos eventos coletados conforme cada evento é recebido em um intervalo de tempo. É uma regra com preservação de estado.

Visão Geral

O padrão de computação executa uma função de computação, conforme definido pelo elemento `<computeFunction>` na linguagem de regra, em relação a cada evento aceito em um período de tempo. O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>`.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de computação, a resposta da regra é executada quando o espaço de tempo é concluído, conforme definido pelo elemento `<onTimeWindowComplete>`. O valor da computação está disponível durante a ação `<onTimeWindowComplete>`.

Exemplo de Uso desse Padrão de Regra

Suponha que um aplicativo esteja processando eventos de pedidos de clientes. Um exemplo de uso do padrão de computação é uma regra que faz o seguinte:

Sempre que um evento é recebido, seu valor total é incluído no valor total de todos os pedidos ocorridos durante o período de tempo especificado e o valor total atualizado de todos os pedidos é publicado em uma interface com o usuário.

Referências relacionadas

“Resumo da Regra de Computação” na página 50

Este resumo lista todos os elementos da linguagem da regra de computação.

Padrão de Duplicata

Uma regra de duplicata é definida pelo padrão de duplicata. Ela conta o segundo e os subseqüentes eventos que são aceitos no intervalo de tempo especificado, mas ignora o processamento do conjunto de regras deles. É uma regra com preservação de estado.

Visão Geral

O padrão de duplicata é utilizado normalmente para isolar eventos semelhantes (duplicatas) em um período de tempo. Um evento em duplicata é semelhante de alguma forma a um evento anterior, mas não é necessariamente uma cópia exata desse evento. Eventos serão duplicatas apenas se atenderem aos critérios de seleção de eventos da regra. O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>` na linguagem da regra.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de duplicata, a resposta da regra é executada nas seguintes ocasiões:

- Quando o primeiro evento for detectado, conforme definido pelo elemento `<onDetection>`.
- Conforme cada evento em duplicata é processado, conforme definido pelo elemento `<onNextEvent>`.
- Quando o espaço de tempo é concluído, conforme definido pelo elemento `<onTimeWindowComplete>`.

O primeiro evento aciona a ação `<onDetection>` embora nenhum evento em duplicata tenha sido recebido. O motivo desse comportamento é que talvez você queira redirecionar o primeiro evento e ignorar o processamento do conjunto de regras de eventos em duplicata. Nesse caso, é possível incluir uma ação de resposta da regra que redirecione o primeiro evento quando a ação `<onDetection>` for acionada para a regra.

O processamento padrão para eventos em duplicata (o segundo e subseqüentes eventos) é contar um evento em duplicata, mas ignorar o processamento do conjunto de regras de um evento em duplicata. Para executar ações adicionais em um evento em duplicata, é possível definir explicitamente uma ação `<onNextEvent>`. Por exemplo, em determinados casos, o evento em duplicata representa um evento que já pode estar registrado em um banco de dados ou em outro repositório. Portanto, talvez você queira codificar uma ação `<onNextEvent>` para remover o evento em duplicata desses outros locais.

Uma ação `<onTimeWindowComplete>` pode ser utilizada para criar um registro de resumo para todos os eventos em duplicata que incluem o número de duplicatas processadas.

Exemplo de Uso desse Padrão de Regra

Suponha que uma mensagem “Negação de Serviço” continue a ocorrer a partir do mesmo tipo de recurso (um monitor de segurança). Isso indica uma possível violação de segurança. Um exemplo de uso do padrão de duplicata é uma regra que faz o seguinte:

Depois de uma mensagem “Negação de Serviço” ocorrer a partir do monitor de segurança, as duplicatas desse evento que ocorrem durante um período de tempo de 30 segundos são contadas, mas não são enviados para o console do operador. Além disso, no final do período de tempo de 30 segundos, a regra gera um evento que indica o número de mensagens “Negação de Serviço” ocorridas em um período de tempo.

Referências relacionadas

“Resumo da Regra de Duplicata” na página 51

Este resumo lista todos os elementos da linguagem da regra de duplicação.

Padrão de Filtragem

Uma regra de filtragem é definida pelo padrão de filtragem. Ela executa uma determinada ação ao aceitar um evento. Ela age apenas em um único evento e é, portanto, uma regra sem preservação de estado.

Visão Geral

O padrão de filtragem é utilizado para agir em eventos individuais que atendem aos critérios de seleção de eventos. Ao contrário de outros padrões de regras, ele não retém informações de estado associadas (como o histórico de eventos passados).

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de filtragem, a resposta da regra é executada quando qualquer evento que atenda aos critérios de seleção de eventos é recebido, conforme definido pelo elemento `<onDetection>`.

Exemplo de Uso desse Padrão de Regra

Um exemplo de uso do padrão de filtragem é uma regra que faz o seguinte:

Se um evento `ServerStatus` indicar um `serverLoad` maior que 95%, a regra executará uma ação que solicitará um administrador por meio de pager.

Referências relacionadas

“Resumo da Regra de Filtragem” na página 52

Este resumo lista todos os elementos da linguagem da regra de filtragem.

Padrão de Seqüência

Uma regra de seqüência é definida pelo padrão de seqüência. Ela detecta se uma determinada seqüência de eventos chega em um intervalo de tempo. A seqüência pode ser ordenada ou aleatória. Uma regra de seqüência é uma regra com preservação de estado.

Visão Geral

O padrão de seqüência procura uma seqüência de eventos em um período de tempo e detecta se a seqüência está completa ou incompleta. Uma seqüência incompleta é aquela que inclui um ou mais dos eventos na ordem especificada, mas não todos.

O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>` na linguagem da regra. Cada evento na seqüência é definido por um elemento `<eventSelector>` separado na regra. A seqüência de eventos pode ser detectada em uma das seguintes ordens:

- Na ordem em que os elementos `<eventSelector>` são codificados para a regra. Nesse caso, quando a regra detecta o evento definido pelo primeiro elemento `<eventSelector>`, a detecção da seqüência é iniciada. Agora, a regra espera o evento definido pelo segundo elemento `<eventSelector>`.
- Na ordem aleatória. Nesse caso, quando a regra detecta um dos eventos definidos pelos elementos `<eventSelector>`, a detecção da seqüência é iniciada. Agora, a regra espera outro dos eventos definidos pelos elementos `<eventSelector>`.

O padrão de seqüência é diferente de outros padrões de regra nas maneiras primárias a seguir:

- Ele possui vários elementos `<eventSelector>` para definir quais eventos são aceitos pela regra. Ele requer um mínimo de dois elementos `<eventSelector>`.

- Quando um evento atende aos critérios definidos por um dos elementos `<eventSelector>`, esse elemento `<eventSelector>` é excluído do processamento adicional de eventos nessa instância de regra.
- O atributo de alias no elemento `<eventSelector>` é válido apenas em uma regra de seqüência e nomeia exclusivamente um evento que é selecionado por um determinado seletor de eventos na regra de seqüência. Numa expressão dentro de um predicado ou ação de filtragem, é possível utilizar a variável `act_eventList` para acessar um evento numa regra de seqüência pelo seu nome de alias.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de seqüência, a resposta da regra é executada nas seguintes ocasiões:

- Quando a seqüência completa de eventos é detectada no espaço de tempo, conforme definido pelo elemento `<onDetection>`.
- Quando um ou mais eventos chegam, mas a seqüência completa não chega dentro do espaço de tempo, conforme definido pelo elemento `<onTimeout>`.
O padrão de seqüência pode ser útil para detectar que uma determinada seqüência está incompleta. Por exemplo, se ocorrer um evento “sistema inativo” sem um evento “sistema ativo” subsequente, o autor de regras poderá codificar uma ação `<onTimeout>` para manipular esse tipo de evento ausente.

Exemplo de Uso desse Padrão de Regra

Cenário que Ilustra a Detecção de Seqüência Completa:

Suponha que em um ambiente de TI, um administrador queira saber se o valor do tamanho de heap do DB2 está afetando o WebSphere Application Server e, nesse caso, corrigir o problema. Então, se os seguintes eventos ocorrerem na ordem a seguir durante um período de tempo especificado, o administrador desejará aumentar o valor do tamanho de heap do DB2 e reiniciar o gerenciador do banco de dados:

1. Uma exceção de alocação de recurso do WebSphere Application Server.
Suponha um evento do tipo `WASResourceAllocationException`.
2. A mensagem de erro do DB2 que informa: “Sem heap suficiente para processar a instrução”. Suponha um evento do tipo `DB2NotEnoughHeap`.

Neste cenário, dois elementos `<eventSelector>` estão definidos numa regra de seqüência e os eventos devem acontecer na ordem em que os elementos `<eventSelector>` estão codificados (ao invés de numa ordem aleatória). O primeiro elemento `<eventSelector>` está procurando o evento `WASResourceAllocationException`; o segundo elemento `<eventSelector>`, o evento `DB2NotEnoughHeap`. Suponha que os eventos a seguir sejam apresentados ao sistema no espaço de tempo especificado:

1. `WASResourceAllocationException`
2. `DB2BackupStarted`
3. `WASResourceAllocationException`
4. `WASResourceAllocationException`
5. `DB2NotEnoughHeap`

O comportamento da regra é o seguinte:

1. O primeiro evento, `WASResourceAllocationException`, será aceito. Como os critérios para o primeiro elemento `<eventSelector>` foram atendidos, o primeiro elemento `<eventSelector>` é agora excluído do processamento adicional de eventos nessa regra.
2. O segundo evento, `DB2BackupStarted`, será ignorado.
3. O terceiro evento, `WASResourceAllocationException`, será ignorado.
4. O quarto evento, `WASResourceAllocationException`, será ignorado.
5. O quinto evento, `DB2NotEnoughHeap`, será aceito e isso conclui a seqüência. A ação de resposta da regra `<onDetection>` será executada. Essa ação será definida para aumentar o valor do tamanho de heap do DB2 e reiniciar o gerenciador de banco de dados. A regra retorna a seu estado inicial.
O primeiro elemento `<eventSelector>` é agora incluído no processamento futuro de eventos por essa regra.

Cenário que Ilustra a Detecção de Seqüência Incompleta:

Suponha que uma organização de negócios pretenda que todos os pedidos dos clientes estejam prontos para entrega em uma hora da entrada do pedido e queira saber quando isso não ocorre.

Neste cenário, dois elementos `<eventSelector>` estão definidos numa regra de seqüência e os eventos devem acontecer na ordem em que os elementos `<eventSelector>` estão codificados (ao invés de numa ordem aleatória). O primeiro elemento `<eventSelector>` está procurando o evento Netsales com `operationType=Order` e o segundo elemento `<eventSelector>` está procurando o evento Netsales com `operationType=Delivery`. Suponha que os eventos a seguir sejam apresentados ao sistema no espaço de tempo especificado de 1 hora:

1. Um evento Netsales com `operationType=Order`
2. Um evento Netsales com `operationType=Order`

O comportamento da regra é o seguinte:

1. O primeiro evento será aceito. Como os critérios para o primeiro elemento `<eventSelector>` foram atendidos, o primeiro elemento `<eventSelector>` é agora excluído do processamento adicional de eventos nessa regra.
2. O segundo evento será ignorado.
3. Como um evento de Netsales com o `operationType=Delivery` não foi recebido dentro do intervalo de tempo especificado, a ação de resposta da regra `<onTimeOut>` será executada. Essa ação será definida para notificar um gerente do negócio que um pedido do cliente não está pronto para entrega em uma hora da entrada do pedido. A regra retorna a seu estado inicial.

O primeiro elemento `<eventSelector>` é agora incluído no processamento futuro de eventos por essa regra.

Conceitos relacionados

“Acessando Informações Relacionadas a Eventos” na página 23

Os exemplos a seguir indicam como acessar informações relacionadas ao evento por meio de variáveis fornecidas pela Tecnologia ACT.

Referências relacionadas

“Resumo da Regra de Seqüência” na página 53

Este resumo lista todos os elementos da linguagem da regra de seqüência.

Padrão de Limite

Uma regra de limite é definida pelo padrão de limite. Ela coleta um grupo de eventos selecionados em um intervalo de tempo e determina, após a recepção de cada evento, se uma condição de limite foi atendida. É uma regra com preservação de estado.

Visão Geral

O padrão de limite coleta eventos em um período de tempo até que um valor de limite seja atendido. O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>` na linguagem da regra.

O padrão de limite fornece as três opções a seguir para um tipo de limite:

limite de contagem de eventos

Esse tipo de limite permite definir o número de eventos que devem atender aos critérios de seleção de eventos em um determinado período de tempo. O valor do limite definido é comparado ao número de eventos aceitos. Quando a contagem de eventos equivaler ao limite definido no espaço de tempo, o limite foi atendido.

Esse tipo de limite pode ser útil para uma verificação de contagem de eventos muito simples. Por exemplo, ele pode responder à pergunta: “Ocorreram 5 eventos de falha de login em 1 minuto?”

Esse limite é definido pelo elemento `<eventCountThreshold>`. O elemento `<eventCountThreshold>` também especifica um dos dois possíveis modos de intervalo de tempo a seguir para o espaço de tempo:

intervalo fixo

Esse intervalo começa quando o primeiro evento que atende aos critérios de seleção de eventos é recebido e termina quando ocorre uma das seguintes situações:

- A regra atende a seu limite na duração de tempo especificada.
- A duração de tempo especificada passou.

intervalo móvel

Esse intervalo começa quando o primeiro evento que atende aos critérios de seleção de eventos é recebido. No entanto, quando a regra não atende a seu limite e a duração de tempo especificada passa, o espaço de tempo ajusta (move) a hora de início para a hora de recepção do evento de um novo “primeiro” evento, que normalmente é o próximo evento que for aceito. O intervalo móvel continua a ser ajustado assim até que ocorra uma das seguintes situações:

- A regra atende a seu limite na duração de tempo especificada.
- Após a recepção do evento que começa o espaço de tempo, nenhum evento subsequente é recebido durante o tempo especificado.

O evento que começa o espaço de tempo (torna-se o novo “primeiro” evento) é aquele com uma hora de recepção que atende a estes critérios: a hora da recepção, somada à duração do intervalo de tempo da regra, é maior que a hora atual. A seguir, os critérios na forma de uma equação:

hora de recepção do evento + duração do intervalo de tempo da regra > hora atual

Quando esse evento não existir, o intervalo móvel não poderá mais ajustar a hora e o intervalo será encerrado.

limite de computação

Esse tipo de limite permite gravar código (ou utilizar código gravado por outra pessoa) que executa um cálculo em cada evento aceito e retorna um valor de limite calculado contido em uma variável definida anteriormente. Esse valor de limite calculado é então comparado a um valor de limite definido para determinar se o limite foi atendido.

Portanto, é possível aplicar um cálculo complexo para criar (ou atualizar) um valor de limite calculado, possivelmente utilizando dados salvos de eventos anteriores, e o autor de regras pode configurar o valor de limite definido independentemente da lógica que calcula o valor de limite calculado.

Esse tipo de limite pode ser útil para a agregação e comparação de um valor com um valor de limite definido. Por exemplo, ele pode ser utilizado para calcular a soma do valor de vendas em dólar para um determinado cliente em um determinado período de tempo e para comparar essa soma com um valor de limite definido.

Esse limite é definido pelo elemento `<computedThreshold>`.

limite booleano

Esse tipo de limite permite gravar código (ou utilizar código gravado por outra pessoa) que retorna um valor de `true` ou `false` para cada evento aceito. Se o valor for `true`, o limite foi atendido. Se o valor for `false`, a regra de limite continuará a ser processada até que o período de tempo termine ou até que ele aceite outro evento.

Esse tipo de limite pode ser útil para verificação de um intervalo de valores. Por exemplo, se a utilização da CPU tiver que ficar sempre entre 30% e 80%, esse limite poderá verificar constantemente se a utilização permanece dentro desse intervalo.

Esse limite é definido pelo elemento `<booleanThreshold>`.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de limite, a resposta da regra é executada nas seguintes ocasiões:

- Quando o limite é atendido, conforme definido pelo elemento `<onDetection>`.
- Quando um ou mais eventos são aceitos, mas o limite não é atendido dentro do espaço de tempo, conforme definido pelo elemento `<onTimeOut>`.

Exemplo de Uso desse Padrão de Regra

Um exemplo de uso do padrão de limite com o limite de contagem de eventos é uma regra que faz o seguinte:

Se mais de 4 eventos `Servidor inatingível` forem originados da mesma sub-rede em um intervalo de tempo móvel de 30 segundos, a regra executará uma ação para verificar o status de um roteador.

Referências relacionadas

“Resumo da Regra de Limite” na página 55

Este resumo lista todos os elementos da linguagem da regra de limite.

Padrão de Cronômetro

Uma regra de cronômetro é definida pelo padrão de cronômetro. Ele inicia ações em intervalos regulares. É uma regra com preservação de estado. Embora uma regra do cronômetro não processe eventos, ela pode ser ativada ou desativada por um evento.

Visão Geral

O padrão de cronômetro é análogo a um cronômetro que inicia no começo de um período de tempo e pára no final do período de tempo. O período de tempo é indicado por um espaço de tempo obrigatório, conforme definido pelo elemento `<timeWindow>` na linguagem da regra.

A não ser que especificado para não repetir, o padrão de cronômetro repetirá até que a regra de cronômetro seja desativada. Portanto, quando a regra de cronômetro for iniciada, ela aguardará o período de tempo especificado antes de iniciar qualquer ação e repetirá esse comportamento até que seja desativada ou que o mecanismo da Tecnologia ACT seja encerrada.

A regra do cronômetro é exclusiva, pois não contém critérios de seleção de eventos. A regra do cronômetro inicia o processamento de acordo com o intervalo de ativação da regra, conforme definido pelo elemento `<activationInterval>`. Se o elemento `<activationInterval>` padrão for utilizado e o padrão de cronômetro for definido para repetir, a regra do cronômetro será iniciada quando for carregada pelo mecanismo da Tecnologia ACT e será parada quando o mecanismo da Tecnologia ACT for encerrado. Para ativar uma regra de cronômetro com um evento, é necessário especificar o evento no elemento `<activateOnEvent>` contido no elemento `<activationInterval>` da regra.

Condições sob as quais a Resposta da Regra é Executada

Com o padrão de cronômetro, a resposta da regra é executada quando o espaço de tempo é concluído, conforme definido pelo elemento `<onTimeWindowComplete>`.

Exemplo de Uso desse Padrão de Regra

O padrão de cronômetro pode ser útil para implementar as regras de limpeza. Um exemplo de uso do padrão de cronômetro é uma regra que faz o seguinte:

A cada 30 minutos, a regra executa uma ação que limpa eventos inofensivos e informativos abertos por um período superior a 48 horas.

Referências relacionadas

“Resumo da Regra de Cronômetro” na página 56

Este resumo lista todos os elementos da linguagem da regra de cronômetro.

Aspectos Comuns e Exclusivos de Padrões de Regras Diferentes

Esta matriz fornece uma visão geral de alto nível dos aspectos comuns e exclusivos dos diferentes padrões de regras.

Tabela 2 na página 20 lista os elementos de linguagem primários das regras e mostra um X na coluna de cada tipo de regra sob a qual o elemento é válido. Os elementos de linguagem primários são os elementos-filhos diretos dos diferentes tipos de regras. A lista não inclui os elementos contidos nesses elementos-filhos

diretos, podendo também variar, dependendo do tipo de regra. Além disso, a validade de determinados atributos de elementos pode variar, dependendo do tipo de regra.

Tabela 2. Matriz Mostrando os Aspectos Comuns e Exclusivos de Padrões de Regras Diferentes

Elemento	coleta	computação	duplicata	filtragem	seqüência	limite	cronômetro
<comment>	X	X	X	X	X	X	X
<variable>	X	X	X	X	X	X	X
<activationInterval>	X	X	X	X	X	X	X
<lifeCycleActions>	X	X	X	X	X	X	X
<eventSelector>	X	X	X	X	X	X	
<groupingKey>	X	X	X		X	X	
<timeWindow>	X	X	X		X	X	X
<computeFunction>		X					
<booleanThreshold>						X	
<computedThreshold>						X	
<eventCountThreshold>						X	
<onDetection>			X	X	X	X	
<onNextEvent>			X				
<onTimeOut>					X	X	
<onTimeWindowComplete>	X	X	X				X

Expressões

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Os autores de regras podem codificar expressões com fins diferentes, dependendo do contexto e do resultado que desejam obter. As expressões são utilizadas com frequência para a inicialização de variáveis, a definição de critérios de seleção de eventos e a especificação de ações de resposta da regra e de ciclo de vida.

Elementos de Linguagem que Contêm Expressões

Cada elementos de linguagem que contém uma expressão tem um atributo `expressionLanguage` que identifica a linguagem de programação na qual a expressão é gravada. A linguagem de programação Java é a única linguagem de expressão suportada.

As expressões podem ser obtidas nos elementos de linguagem de regra a seguir.

- `<varInitializer>` para um conjunto de regras, bloco de regra ou variável de regra
- `<filteringPredicate>` em `<eventSelector>`
- `<computedValue>` em `<groupingKey>`
- `<computeFunction>` em uma regra de computação
- `<booleanThreshold>` em uma regra de limite
- `<computedThreshold>` em uma regra de limite
- Ações de resposta de uma regra:
 - `<action>` em `<onDetection>`. Esta ação será válida apenas para as regras de duplicata, de filtragem, de seqüência e de limite.

- <action> em <onNextEvent>. Esta ação é válida apenas para a regra de duplicata.
- <action> em <onTimeOut>. Esta ação é válida apenas para as regras de seqüência e de limite.
- <action> em <onTimeWindowComplete>. Esta ação é válida apenas para as regras de coleta, de computação, de duplicata e de cronômetro.
- Ações de ciclo de vida de uma regra:
 - <action> em <onLoad>
 - <action> em <onActivation>
 - <action> em <onDeactivation>
 - <action> em <onUnload>

O que a Tecnologia ACT Fornece para Ajudar nas Codificação de Expressões

Para ajudar o autor de regras na codificação de expressões, a Tecnologia ACT fornece recursos para fazer o seguinte:

- Importar módulos externos (por exemplo, classes Java) e objetos para serem utilizados nas expressões.
- Inicializar e acessar o conjunto de regras, o bloco de regra ou as variáveis de regra.
- Por meio da variável `act_event`, acessar o evento atual que está sendo processado por uma regra.
- Por meio da variável `act_eventCount`, acessar o número de eventos aceitos por uma regra.
- Por meio da variável `act_eventList`, acessar a lista de eventos aceitos por uma regra. Isso inclui o recurso para acessar vários atributos de um evento e para acessar cada evento em uma regra de seqüência pelo seu nome de alias.
- Por meio da variável `act_lib`, acessar métodos que incluem o recurso de obter e configurar variáveis e para controlar o fluxo de eventos por meio de um conjunto de regras.
- Por meio da variável `act_location`, acessar o local, na hierarquia da regra, de uma expressão.
- Por meio da variável `act_nodeName`, acessar o nome completo de um nó.
- Por meio da variável `act_threshold`, acessar o valor do limite definido de uma regra de limite.

Importando e Acessando Módulos e Objetos Externos

Este exemplo indica como tornar código externo (por exemplo, classes Java) e objetos externos acessíveis às expressões. Objeto externo é aquele que o aplicativo cria para se comunicar com as expressões.

Antes de poder acessar o código externo de uma expressão, é necessário torná-lo acessível às expressões.

Uma importação é uma maneira específica da linguagem de programação tornar código externo acessível a expressões. O elemento `<import>` contém um tipo especial de expressão que especifica os módulos externos (por exemplo, classes Java) a serem importados para utilização em outras expressões nas regras. Uma importação pode ser definida no nível do conjunto de regras ou do bloco de regra.

O elemento `<import>` a seguir contém uma expressão, criada na linguagem de programação Java, importando as classes `StaticHelper` e `Queue`, que pode ser referida por outras expressões:

```
<import expressionLanguage="java">
  import com.ibm.act.sample.StaticHelper;
  import com.ibm.act.test.Queue;
</import>
```

Embora o uso do nome completo da classe não seja necessário na instrução de importação, é necessário especificar o nome completo para evitar um tempo longo de compilação. Por exemplo, a classe Java deve ser especificada como `com.ibm.act.sample.StaticHelper` em vez de `com.ibm.act.sample.*` ou `com.ibm.act.*`.

Acessando um Método Estático

O exemplo a seguir indica como uma expressão em uma ação de resposta da regra referencia a classe `StaticHelper` após a importação da classe:

```
<onDetection>
  <action expressionLanguage="java">
    StaticHelper.pageAdministrator("Too many login attempts for " + act_event.getAttribute("userID"));
  </action>
</onDetection>
```

Acessando um Método de Instância de um Objeto

O exemplo a seguir indica como uma expressão em uma ação de resposta da regra faz referência à classe `Queue` após a importação da classe. Nesse exemplo, um objeto interno com nome `OutputQueueOne` e tipo `Queue` é obtido e utilizado para colocar um evento numa fila específica.

```
<onDetection>
  <action expressionLanguage="java">
    Queue myQueue = (Queue)act_lib.getExternalContext("OutputQueueOne");
    myQueue.enqueue(act_event);
  </action>
</onDetection>
```

Inicializando e Acessando Variáveis

Este exemplo indica como inicializar e acessar o conjunto de regras, o bloco ou as variáveis de regra.

Uma variável pode ser definida no nível do conjunto de regras, de um bloco de regra ou de uma regra. Para que uma variável possa ser acessada, ela deve ser inicializada com uma expressão de inicialização. A expressão a seguir inicializa duas variáveis, uma chamada `hostsList` e outra chamada `hostsString`:

```
<variable name="hostsList" dataType="java.util.ArrayList">
  <varInitializer expressionLanguage="java">
    return new ArrayList();
  </varInitializer>
</variable>
<variable name="hostsString" dataType="java.lang.String">
  <varInitializer expressionLanguage="java">
    return new String();
  </varInitializer>
</variable>
```

Todas as variáveis são acessadas por meio de expressões. O exemplo a seguir mostra como as variáveis `hostsList` e `hostsString`, inicializadas no exemplo anterior, são acessados por meio de uma expressão em uma ação de resposta da regra. Neste exemplo, `hostsList` é modificado e `hostsString` recebe um novo valor.

```

<onNextEvent>
  <action expressionLanguage="java">
    String hostname = act_event.getStringAttribute("hostname");
    ArrayList hostsList = (ArrayList)act_lib.getVariable("hostsList");
    hostsList.add(hostname);
    String hostsString = act_lib.getStringVariable("hostsString");
    String newHostString = hostsString + ", " + hostname;
    act_lib.setStringVariable("hostsString", newHostsString);
  </action>
</onNextEvent>

```

Acessando Informações Relacionadas a Eventos

Os exemplos a seguir indicam como acessar informações relacionadas ao evento por meio de variáveis fornecidas pela Tecnologia ACT.

Exemplo de Acesso ao Evento Atual:

O código a seguir mostra como utilizar a variável `act_event` para obter o atributo nome do host de um evento:

```
act_event.getAttribute("hostname");
```

Exemplo de Acesso a Eventos por meio da Lista de Eventos pelo Índice:

O código a seguir mostra como utilizar a variável `act_eventList` para obter o primeiro evento na lista de eventos:

```
act_eventList.get(0);
```

Exemplo de Acesso a Eventos por meio da Lista de Eventos pelo Alias:

Ao contrário de outros tipos de regras, a regra de sequência permite vários seletores de eventos e requer realmente um mínimo de dois seletores de eventos. O atributo de alias no elemento `<eventSelector>` é válido apenas em uma regra de sequência e nomeia exclusivamente um evento que é selecionado por um determinado seletor de eventos na regra de sequência. Numa expressão dentro de um predicado ou ação de filtragem, é possível utilizar a variável `act_eventList` para acessar um evento numa regra de sequência pelo seu nome de alias.

O código a seguir mostra dois seletores de eventos para uma regra de sequência. Os nomes do alias `TECevent` e `WASevent`.

```

<eventSelector alias="TECevent">
  <eventType type="serverStatus"/>
  <filteringPredicate expressionLanguage="java">
    return act_event.getStringAttribute("source").equals("TEC");
  </filteringPredicate>
</eventSelector>
<eventSelector alias="WASevent">
  <eventType type="serverStatus"/>
  <filteringPredicate expressionLanguage="java">
    return act_event.getStringAttribute("source").equals("WAS");
  </filteringPredicate>
</eventSelector>

```

O código a seguir mostra como utilizar a variável `act_eventList` para obter o evento aceito pelo primeiro seletor de evento denominado `TECevent`:

```
act_eventList.get("TECevent");
```

Boas Práticas para Codificação de Expressões

Estas informações incluem algumas boas práticas, dicas e sugestões para codificar expressões eficaz e efetivamente.

- Para facilitar o entendimento, a maioria dos exemplos de expressões fornecidos nestas informações inclui o código Java diretamente nos constructos XML. No entanto, quando você cria regras, a melhor prática é utilizar módulos externos para conter o código Java e chamar tais módulos como parte das expressões.

Também é possível utilizar ou editar os fragmentos existentes (ou criar novos fragmentos) no construtor de regras para fornecer o código para as chamadas aos módulos externos. Fragmentos são seleções de código fonte que podem ser utilizados em expressões. No construtor de regras, os fragmentos estão disponíveis na visualização Fragmentos.

Nessa abordagem, as tarefas de projetar, desenvolver, editar, testar e depurar código Java podem ser feitas no IDE (ambiente de desenvolvimento integrado) de sua preferência e controladas como uma parte regular do processo de desenvolvimento.

- Para evitar que o código de expressão seja analisado como XML, coloque o código em uma seção CDATA, em que `<![CDATA[` anteceda imediatamente o código e `]]>` siga imediatamente o código, conforme mostrado no seguinte exemplo:

```
<onTimeout>
<action expressionLanguage="java">
<![CDATA[
    IEvent firstEvent = act_eventList.get(0);
    System.out.println("Expired Item: " + firstEvent.getAttribute("sourceComponentId.location"));
]]>
</action>
</onTimeout>
```

Os analisadores XML ignoram tudo que estiver dentro de uma seção CDATA.

- Se utilizar o método `act_lib.getExternalContext()` numa expressão, não armazene o objeto retornado a partir do método num conjunto de regras ou variável de bloco de regra. O motivo é que um aplicativo pode alterar a referência à instância do objeto, e o conjunto de regras associado ou bloco de regra disponível não está atualizado.
- Se você utilizar uma instrução de retorno (`return;`) em uma expressão em um elemento `<action>` e elementos `<action>` adicionais forem codificados para a respectiva ação de resposta de regra ou de ciclo de vida, a execução do código terminará onde a instrução de retorno estiver codificada e começará na expressão no próximo elemento `<action>`.
- O gerenciamento de regras e outros métodos do mecanismo de Tecnologia ACT não poderão ser chamados a partir de uma ação de resposta de regra ou de ciclo de vida.

Informações Relacionadas

“Importando e Acessando Módulos e Objetos Externos” na página 21

Este exemplo indica como tornar código externo (por exemplo, classes Java) e objetos externos acessíveis às expressões. Objeto externo é aquele que o aplicativo cria para se comunicar com as expressões.

“Incluindo Fragmentos em Expressões Dentro das Regras” na página 43

É possível incluir, em expressões dentro das regras, os fragmentos da visualização Fragmentos do Eclipse Workbench.

Variáveis

Na linguagem de regras, determinadas variáveis são utilizadas para armazenar informações relacionadas a eventos nas diferentes ocorrências ou regras do evento. Essas informações relacionadas a eventos poderão, então, ser acessadas a partir de expressões nas regras. Alguns tipos de variáveis são definidos pelo autor de regras

e outros são fornecidos pela Tecnologia ACT. Alguns tipos podem ser acessados diretamente de uma expressão e outros podem ser acessados apenas por meio de métodos fornecidos pela Tecnologia ACT.

Variáveis Definidas nos Elementos <variable> e Acessados por Meio de Métodos

O usuário pode definir uma variável dentro do elemento <variable> de uma regra, bloco de regra ou conjunto de regras. É possível acessar essa variável em uma expressão, utilizando um dos seguintes métodos:

- O método `getVariable()` ou um dos métodos `getjavalatypeVariable()`
- O método `setVariable()` ou um dos métodos `setjavalatypeVariable()`

Por exemplo, se definir a variável `rule_writer_variable` no elemento <variable> de uma regra, será possível acessar tal variável com o seguinte código:

```
int sample_variable = act_lib.getIntVariable("rule_writer_variable");
```

Variáveis Fornecidas pela Tecnologia ACT e Acessadas Diretamente numa Expressão

As variáveis a seguir são fornecidas pela Tecnologia ACT. É possível utilizar variáveis sequenciais dentro de uma expressão.

- `act_event`
- `act_eventList`
- `act_lib`

Por exemplo, com o código a seguir, é possível acessar a variável `act_event` para obter o atributo `hostname` de um evento:

```
act_event.getAttribute("hostname");
```

Variáveis Fornecidas pela Tecnologia ACT e Acessadas por Meio de Métodos

As variáveis a seguir são fornecidas pela Tecnologia ACT. É possível acessar as variáveis numa expressão por meio do método `getVariable()` ou de um dos métodos `getjavalatypeVariable()`.

- `act_eventCount`
- `act_location`
- `act_nodeName`
- `act_threshold`

Por exemplo, com o código a seguir, é possível acessar a variável `act_eventCount`:

```
int eventcount_integer = act_lib.getIntVariable(IACTLibrary.EVENTCOUNT);
```

A Tabela 3 na página 26 apresenta as constantes que a interface `IACTLibrary` fornece para essas variáveis. No seu código, para garantir que erros de ortografia ou tipográficos sejam descobertos na compilação em vez de no tempo de execução, sempre utilize as constantes que representam tais variáveis em vez das variáveis em si. Por exemplo, utilize `act_lib.getIntVariable(IACTLibrary.EVENTCOUNT)`; em vez de `act_lib.getIntVariable("act_eventCount")`;

Tabela 3. Variáveis com Constantes Associadas

Variável	Constante Associada
act_eventCount	EVENTCOUNT
act_location	LOCATION
act_nodeName	NODENAME
act_threshold	THRESHOLD

Referências relacionadas

“Elemento variable” na página 115

O elemento <variable> define uma variável e contém informações em uma forma que pode ser referenciada por expressões. Uma variável pode ser definida no nível do conjunto de regras, de um bloco de regra ou de uma regra.

Tipos de Dados das Variáveis da Tecnologia ACT

As variáveis fornecidas pela Tecnologia ACT têm diferentes tipos de dados.

A Tabela 4 apresenta os tipos de dados dessas variáveis.

Tabela 4. Tipos de Dados das Variáveis da Tecnologia ACT

Variável	Tipo de dado
act_event	O tipo definido pela interface com.ibm.correlation.IEvent
act_eventCount	int
act_eventList	O tipo definido pela interface com.ibm.correlation.IEventList
act_lib	O tipo definido pela interface com.ibm.correlation.IACTLibrary
act_location	java.lang.String
act_nodeName	java.lang.String
act_threshold	Como essa variável é o valor do atributo threshold no elemento <computedThreshold> ou <eventCountThreshold> de uma regra de limite, o tipo de dados deve ser idêntico ao do valor do atributo threshold.

Contextos de Expressões nos quais as Variáveis são Válidas

As variáveis fornecidas pela Tecnologia ACT são válidas apenas em contextos específicos da expressão.

A Tabela 5 lista os contextos de expressão nas quais essas variáveis são válidas. A tabela mostrará um X na coluna de cada variável que for válida no respectivo contexto de expressão. Restrições adicionais de uso que se aplicam a essas variáveis são descritas no tópico da respectiva variável.

Tabela 5. Contextos de Expressões nos quais as Variáveis são Válidas

Contexto da expressão	act_event	act_eventCount	act_eventList	act_lib	act_location	act_nodeName	act_threshold
<action> em <onActivation>	X			X	X	X	X
<action> em <onDeactivation>	X	X	X	X	X	X	X
<action> em <onDetection>	X	X	X	X	X	X	X
<action> em <onLoad>				X	X	X	X
<action> em <onNextEvent>	X	X	X	X	X	X	
<action> em <onTimeOut>		X	X	X	X	X	X
<action> em <onTimeWindowComplete>		X	X	X	X	X	

Tabela 5. Contextos de Expressões nos quais as Variáveis são Válidas (continuação)

Contexto da expressão	act_event	act_eventCount	act_eventList	act_lib	act_location	act_nodeName	act_threshold
<action> em <onUnload>				X	X	X	X
<booleanThreshold>	X	X	X	X	X	X	
<computedThreshold>	X	X	X	X	X	X	X
<computedValue>	X			X	X	X	
<computeFunction>	X	X	X	X	X	X	
<filteringPredicate>	X	X	X	X	X	X	X
<varInitializer> para uma regra				X	X	X	X

Variável act_event

A variável act_event fornece acesso a métodos que são aplicados ao evento atual.

Detalhes

Como uma regra de cronômetro não processa eventos, a variável act_event em uma regra de cronômetro será aplicada apenas aos eventos que ativarem ou desativarem a regra.

A variável act_event é aplicável dentro das ações <onActivation> e <onDeactivation> apenas se um evento ativar ou desativar a regra. Caso contrário, essa variável é nula.

Exemplo de Código

O código a seguir acessa a variável act_event para obter o atributo hostname de um evento:

```
String host = act_event.getStringAttribute("hostname");
```

Métodos que Podem Ser Acessados

Os métodos aos quais a variável act_event fornece acesso são definidos na interface IEvent, conforme mostrado em Tabela 6 na página 28.

Tabela 6. Interface IEvent com Métodos Correspondentes e Local das Descrições de Métodos Javadoc

Interface	Métodos	Local das descrições de métodos Javadoc
IEvent	<ul style="list-style-type: none"> • get • getAttribute • getBooleanAttribute • getByteAttribute • getShortAttribute • getIntAttribute • getLongAttribute • getFloatAttribute • getDoubleAttribute • getStringAttribute • set • getTimeStamp • setTimeStamp • getType • getOriginal 	com.ibm.correlation.IEvent

Variável act_eventCount

A variável act_eventCount é um inteiro igual ao número de eventos aceitos por uma regra.

Detalhes

Para uma regra de duplicata, o valor da variável act_eventCount é o número total de eventos aceitos, que inclui o evento original e as duplicatas. Para todos os outros tipos de regras, o valor é o mesmo que o tamanho da lista de eventos, que pode ser obtido por meio da variável act_eventList, utilizando o método act_eventList.size().

As variáveis act_eventCount e act_eventList não são válidas em uma regra do cronômetro porque essa regra não processa eventos.

Se uma regra for definida com uma chave de agrupamento, as variáveis act_eventCount, act_eventList e act_threshold não serão válidas nos seguintes contextos de expressão:

- Ações do ciclo de vida
- <filteringPredicate> em <activateOnEvent> ou <deactivateOnEvent> em <activationInterval>
- <computedValue>

Isso ocorre porque, nesse caso, as variáveis da regra são aplicadas somente a uma instância da regra e as instâncias de regras não existem na hora em que essas expressões são executadas.

Exemplo de Código

O código a seguir acessa a variável act_lib para obter o número de eventos aceitos por uma regra:

```
int eventCt = act_lib.getIntVariable(IACTLibrary.EVENTCOUNT);
```

Variável `act_eventList`

A variável `act_eventList` fornece acesso a métodos que são aplicados à lista de eventos aceitos por uma regra.

Detalhes

Uma regra de filtragem e uma regra de duplicata sempre têm uma lista de não mais de um evento porque uma regra de filtragem é uma regra sem preservação de estado e porque uma regra de duplicata retém apenas o primeiro evento analisado.

As variáveis `act_eventCount` e `act_eventList` não são válidas em uma regra do cronômetro porque essa regra não processa eventos.

Se uma regra for definida com uma chave de agrupamento, as variáveis `act_eventCount`, `act_eventList` e `act_threshold` não serão válidas nos seguintes contextos de expressão:

- Ações do ciclo de vida
- `<filteringPredicate>` em `<activateOnEvent>` ou `<deactivateOnEvent>` em `<activationInterval>`
- `<computedValue>`

Isso ocorre porque, nesse caso, as variáveis da regra são aplicadas somente a uma instância da regra e as instâncias de regras não existem na hora em que essas expressões são executadas.

Exemplo de Código

O código a seguir acessa a variável `act_eventList` para obter o segundo evento na lista de eventos:

```
IEvent second_event = act_eventList.get(1);
```

Métodos que Podem Ser Acessados

Os métodos aos quais a variável `act_eventList` fornece acesso são definidos na interface `IEventListener`, conforme mostrado em Tabela 7.

Tabela 7. Interface `IEventListener` com Métodos Correspondentes e Local das Descrições de Métodos Javadoc

Interface	Métodos	Local das descrições de métodos Javadoc
<code>IEventListener</code>	<ul style="list-style-type: none">• <code>get</code>• <code>size</code>• <code>isEmpty</code>• <code>listIterator</code>	<code>com.ibm.correlation.IEventListener</code>

Variável `act_lib`

A variável `act_lib` fornece acesso a métodos de biblioteca na Tecnologia ACT.

Detalhes

Os métodos que a variável `act_lib` pode acessar variam dependendo do elemento de linguagem da regra que contém a expressão em que a variável é utilizada. Consulte Tabela 8 na página 30.

Tabela 8. Métodos que a Variável `act_lib` Pode Acessar com Base no Contexto de sua Expressão Contida

Contexto da expressão	Métodos IACTLibrary	Métodos IExitableActionLibrary	Métodos IActionLibrary
<action> em <onActivation>	X		
<action> em <onDeactivation>	X		
<action> em <onDetection>	X	X	X
<action> em <onLoad>	X		
<action> em <onNextEvent>	X	X	X
<action> em <onTimeOut>	X		X
<action> em <onTimeWindowComplete>	X		X
<action> em <onUnload>	X		
<booleanThreshold>	X		
<computedThreshold>	X		
<computedValue>	X		
<computeFunction>	X		
<filteringPredicate>	X		
<varInitializer>	X		

Exemplo de Código

O código a seguir acessa a variável `act_lib` para chamar o método existente no conjunto de regras:

```
act_lib.exitRuleSet();
```

Métodos que Podem Ser Acessados

Os métodos aos quais a variável `act_lib` fornece acesso são definidos nas interfaces `IACTLibrary`, `IExitableActionLibrary` e `IActionLibrary`, conforme mostrado em Tabela 9 na página 31.

Tabela 9. Interfaces com Métodos Correspondentes e Local das Descrições de Métodos Javadoc

Interface	Métodos	Local das descrições de métodos Javadoc
IACTLibrary	<ul style="list-style-type: none"> • trace • getVariable • getBooleanVariable • getShortVariable • getIntVariable • getLongVariable • getFloatVariable • getDoubleVariable • getStringVariable • setVariable • setBooleanVariable • setShortVariable • setIntVariable • setLongVariable • setFloatVariable • setDoubleVariable • setStringVariable • getExternalContext 	com.ibm.correlation.IACTLibrary
IActionLibrary	<ul style="list-style-type: none"> • forward • forwardOnCompletion • activate • deactivate <p>Os métodos definidos na interface IACTLibrary também estão disponíveis para a interface IActionLibrary.</p>	com.ibm.correlation.IActionLibrary
IExitableActionLibrary	<ul style="list-style-type: none"> • exitRuleSet • exitRuleBlock <p>Os métodos definidos nas interfaces IACTLibrary e IActionLibrary também estão disponíveis para a interface IExitableActionLibrary.</p>	com.ibm.correlation.IExitableActionLibrary

Variável act_location

A variável act_location é uma cadeia que identifica o local, na hierarquia da regra, de uma expressão.

Detalhes

O local é um nome completo que indica a posição da expressão na hierarquia da regra. Ele tem a forma *identifier.identifier....*, em que cada ocorrência de *identifier* é uma das seguintes:

- O valor do atributo de nome de um elemento XML que estiver na hierarquia respectiva.
- Para elementos que ocorrerem várias vezes em um bloco de regra ou em uma regra e não tiverem atributo de nome: o elemento XML que contiver a expressão,

seguida de um número de índice entre colchetes. Esse número de índice indica a posição da expressão em seu elemento contido. O contador para designar números de índice começa com 0, em vez de 1, portanto, se um elemento estiver contido no terceiro elemento <action>, por exemplo, o número do índice será mostrado como `action[2]`.

Esses identificadores estão em ordem decrescente, do bloco de regra do nível mais alto para o elemento de nível mais baixo que contiver a expressão.

Exemplo de Código

O código a seguir acessa a variável `act_lib` para obter o local da expressão:

```
String location = act_lib.getStringVariable(IACTLibrary.LOCATION);
```

Exemplos do Local Retornado da Variável

Os valores a seguir são exemplos do local retornado da variável `act_location`.

`ruleBlockA.ruleA.eventSelector[3].filteringPredicate`

Essa expressão está contida no seguinte:

- O bloco de regra com um valor de atributo de nome de `ruleBlockA`
- A regra com um valor de atributo de nome de `ruleA`
- O quarto elemento <eventSelector>
- O elemento <filteringPredicate>

`ruleBlockA.ruleA.onDetection.action[5]`

Essa expressão está contida no seguinte:

- O bloco de regra com um valor de atributo de nome de `ruleBlockA`
- A regra com um valor de atributo de nome de `ruleA`
- O elemento <onDetection>
- O sexto elemento <action>

`ruleBlockA.ruleA.variableA.varInitializer`

Essa expressão está contida no seguinte:

- O bloco de regra com um valor de atributo de nome de `ruleBlockA`
- A regra com um valor de atributo de nome de `ruleA`
- A variável com um valor de atributo de nome de `variableA`
- O elemento <varInitializer>

Variável `act_nodeName`

A variável `act_nodeName` é uma cadeia que identifica o nome completo de um nó.

Detalhes

Na Tecnologia ACT, um nó é um objeto na hierarquia de regra que pode ser incluído, removido ou substituído individual e independentemente em um conjunto de regras. Especificamente, estes objetos são nós: regras, blocos de regras, variáveis de blocos de regras e variáveis de conjuntos de regras. Como um objeto não pode ser operado individual e independentemente abaixo do nível de regra, uma variável de regra não é um nó.

O nome de nó completo de uma regra com um valor de atributo de nome de `rule1` que está localizado em um bloco de regra com um valor de atributo de nome de `ruleBlockA` é `ruleBlockA.rule1`. Como as regras são organizadas hierarquicamente

em um conjunto de regras, um ponto (.) é utilizado em um nome de nó completo para indicar uma descida até um nó de nível menor.

Exemplo de Código

O código a seguir acessa a variável `act_nodeName` para obter o nome completo de um nó:

```
String nodeName = act_lib.getStringVariable(IACTLibrary.NODENAME);
```

Variável `act_threshold`

A variável `act_threshold` é o valor do atributo `threshold`, definido no valor `threshold`, no elemento `<computedThreshold>` ou `<eventCountThreshold>` de uma regra de limite.

Detalhes

A variável `act_threshold` é válida apenas em uma regra de limite.

Se uma regra for definida com uma chave de agrupamento, as variáveis `act_eventCount`, `act_eventList` e `act_threshold` não serão válidas nos seguintes contextos de expressão:

- Ações do ciclo de vida
- `<filteringPredicate>` em `<activateOnEvent>` ou `<deactivateOnEvent>` em `<activationInterval>`
- `<computedValue>`

Isso ocorre porque, nesse caso, as variáveis da regra são aplicadas somente a uma instância da regra e as instâncias de regras não existem na hora em que essas expressões são executadas.

Exemplo de Código

O código a seguir acessa a variável `act_lib` para obter um valor `threshold` definido:

```
int threshold = act_lib.getIntVariable(IACTLibrary.THRESHOLD);
```

Fluxo de Eventos por meio de um Conjunto de Regras

Os eventos fluem por um conjunto de regras na ordem em que os blocos de regras e as regras são codificados. Quando o mecanismo da Tecnologia ACT recebe um evento, ele determina o tipo de evento e identifica as regras que utilizam esse tipo de evento para ativação da regra, processamento do evento ou desativação da regra.

Como os Eventos São Utilizados pelas Regras

Cada regra que utiliza o evento primeiro, determina se ele atende a todos os critérios especificados para ativação da regra, processamento do evento ou desativação da regra. Se atender, a regra executará as seguintes ações:

Para ativação da regra

As ações no elemento `<onActivation>` da regra serão executadas, se estiverem codificadas.

Para processamento do evento

A regra processa o evento. Quando o padrão de regra for correspondido,

as ações de resposta da regra serão executadas, se estiverem codificadas. Em algumas situações, as ações de resposta da regra podem fazer o seguinte:

- A ação pode fazer com que o evento ignore o processamento no restante do bloco de regra ou do conjunto de regras.
- A ação pode enviar um evento novo ou existente para outra regra ou bloco de regra para processamento.

Para desativação da regra

As ações no elemento `<onDeactivation>` da regra serão executadas, se estiverem codificadas.

Métodos que Podem Influenciar o Fluxo de Eventos

A Tecnologia ACT fornece os métodos a seguir que podem ser chamados para influenciar o fluxo de eventos por meio do conjunto de regras. Estes métodos ficam disponíveis por meio da variável `act_lib`.

exitRuleSet

Esse método especifica que o evento atual não será processado por nenhuma regra adicional no conjunto de regras.

exitRuleBlock

Esse método especifica que o evento atual não será processado por nenhuma regra adicional no bloco de regra atual ou em nenhum bloco de regra que esse bloco de regra contiver. No entanto, ele será processado por regras adicionais fora do escopo do bloco de regra atual.

forward

Esse método especifica que um evento deve ser enviado para outras regras e blocos de regras apesar da regra atual não ter concluído seu processamento. Cada uma das outras regras e blocos de regras processará então o evento completamente antes de retorná-lo para a regra que chamou o método `forward`.

forwardOnCompletion

Esse método especifica que um evento deve ser enviado para outras regras e blocos de regras depois que a regra atual tiver concluído seu processamento.

Capítulo 3. Visão Geral da Gravação de Regras

Antes de gravar regras para correlacionar eventos, é necessário entender e planejar a correlação de eventos e projetar as regras. É possível utilizar o construtor de regras da Tecnologia ACT para gravar as regras e compilar o conjunto de regras.

O construtor de regras da Tecnologia ACT é uma GUI para a gravação de regras. Ele inclui ajuda on-line. No construtor de regras, o arquivo de conjunto de regras resultante é um arquivo XML com um tipo de arquivo de actl.

A Tecnologia ACT não fornece um gerenciamento de mudanças ou um sistema de controle de versão.

Planejamento para a Correlação de Eventos

O planejamento para a correlação de eventos inclui entender ou aprender o que é correlação de eventos e como pode ser aplicada em seu aplicativo.

Assegure-se de que entenda os seguintes conceitos:

- As informações de Capítulo 1, “Introdução”, na página 3 e Capítulo 2, “Visão Geral da Linguagem de Regra”, na página 5
- Os eventos processados por seu aplicativo

Cada aplicativo pode processar um conjunto diferente de eventos, conforme descrito nos seguintes exemplos:

Exemplo de negócios de seguros

Em negócios de seguros, os eventos que acompanham o fluxo de trabalho por meio do processo de indenização pode ser gerado e correlacionado para determinar se os processos de negócios estão sendo concluídos de maneira adequada.

Exemplo de vendas

Em um tipo diferente de negócios, os resultados das vendas podem ser resumidos, relatados e comparados periodicamente a um objetivo para indicar o status de alcance dos objetivos de vendas em um determinado período de tempo.

Exemplo de ambiente de TI

Em um ambiente de TI, um sistema crítico pode gerar um evento a cada minuto para indicar que um servidor de banco de dados está sendo executado normalmente. É possível gravar regras de correlação para monitorar o recebimento desses eventos de pulsação e para executar determinadas ações de resposta da regra se um evento de pulsação esperado não for recebido.

Também seria aconselhável entender o formato dos eventos processados por seu aplicativo. A Tecnologia ACT fornece classes e métodos Java para acessar os dados nos eventos processados pelo mecanismo da Tecnologia ACT. No entanto, é importante ter um entendimento básico dos objetos de eventos de base para utilizar essas classes e métodos para acessar ou alterar os eventos durante seu processamento.

Para planejar a correlação de eventos, execute as seguintes etapas:

1. Determine os eventos de seu aplicativo que deseja correlacionar.
2. Determine os padrões de regra para correlacionar os eventos.

Um padrão de regra representa uma situação específica de correlação de eventos e pode ser utilizado para correlacionar eventos que contribuem de alguma maneira para essa situação. Pense em como os eventos processados por seu aplicativo estão relacionados com os padrões de regra definidos pela linguagem de regra da Tecnologia ACT. Isso pode ajudá-lo a determinar os padrões de regra a serem utilizados.

Utilize sempre o padrão mais apropriado para a sua situação de correlação de eventos. Por exemplo, se quiser que uma regra detecte uma determinada sequência de eventos, não crie código para incluir o comportamento do padrão da sequência nas ações de resposta de uma regra de filtragem. Em vez disso, utilize o padrão de sequência para criar uma regra de sequência.

3. Identifique os constructos de cada padrão de regra que deseja utilizar.

As informações a seguir resumem os constructos primários na linguagem de regra, embora os detalhes de cada um deles sejam exclusivos para o padrão de regra. Essas informações são organizadas aproximadamente da mesma maneira com que são apresentadas pela GUI do construtor de regras:

Características

A definição das características da regra, incluindo o nome da regra, a descrição e o padrão. Para obter detalhes, consulte os seguintes tópicos:

- “Elemento collectionRule” na página 75
- “Elemento computationRule” na página 77
- “Elemento duplicateRule” na página 84
- “Elemento filterRule” na página 91
- “Elemento sequenceRule” na página 106
- “Elemento thresholdRule” na página 111
- “Elemento timerRule” na página 113

Variáveis

A definição das variáveis de regras, incluindo o nome, o tipo, a descrição e a expressão de inicialização de cada uma. Para obter detalhes, consulte “Elemento variable” na página 115.

Seleção de evento

A definição dos critérios que determinam quais eventos são aceitos para processamento pela regra. Para obter detalhes, consulte “Elemento eventSelector” na página 88.

Chave de Agrupamento

A definição da chave de agrupamento, que é a maneira de direcionar a regra para criar uma instância de regra separada (ou uma cópia de si mesma) para cada grupo de eventos que compartilham características comuns. Para obter detalhes, consulte “Elemento groupingKey” na página 93.

Características do padrão

A especificação do período de tempo durante o qual a regra com preservação de estado está sendo processada para corresponder a seu padrão e a definição de aspectos exclusivos de determinados padrões de regras com preservação de estado. Para obter detalhes, consulte “Elemento timeWindow” na página 115.

Para a regra de computação, isso inclui a definição do cálculo a ser aplicado nos eventos coletados. Para obter detalhes, consulte “Elemento computeFunction” na página 81.

Para a regra de limite, isso inclui a definição do tipo de limite e outras informações específicas desse tipo. Para obter detalhes, consulte os seguintes tópicos:

- “Elemento booleanThreshold” na página 74
- “Elemento computedThreshold” na página 78
- “Elemento eventCountThreshold” na página 86

Respostas da regra

A definição das ações a serem executadas quando a regra conclui seu processamento.

Para obter detalhes, consulte os seguintes tópicos:

- Para as regras de duplicata, de filtragem, de sequência ou de limite: “Elemento onDetection” na página 99
- Para as regras de duplicata: “Elemento onNextEvent” na página 100
- Para as regras de sequência ou de limite: “Elemento onTimeOut” na página 101
- Para as regras de coleta, de computação, de duplicata ou do cronômetro: “Elemento onTimeWindowComplete” na página 102

Intervalo de ativação

A definição de quando uma regra está ativa e inativa. Para obter detalhes, consulte “Elemento activationInterval” na página 69.

Ciclo de vida

A definição das ações, se houver, a serem executadas nos quatro estágios primários a seguir no ciclo de vida de uma regra: carregamento, ativação, desativação e descarregamento. Normalmente, essas ações não precisam ser definidas. Para obter detalhes, consulte “Elemento lifeCycleActions” na página 96.

4. Identifique os métodos Java e os fragmentos associados chamados dentro das expressões da regra. Em vez de criar um extensivo código Java dentro das expressões da regra, os seus autores devem utilizar os métodos Java para chamar módulos externos. Esses módulos externos podem ser fornecidos pelo aplicativo que incorpora a Tecnologia ACT ou criados pelo autor da regra, como necessário. Os fragmentos associados a cada um dos métodos Java também devem ser identificados. Para obter mais informações, consulte “Boas Práticas para Codificação de Expressões” na página 23.

Continue com “Projetando as Regras para Correlacionar os Eventos”.

Projetando as Regras para Correlacionar os Eventos

Depois de planejar a correlação de eventos, é necessário projetar as regras para correlacionar os eventos.

Primeiramente, conclua “Planejamento para a Correlação de Eventos” na página 35.

Para projetar as regras, execute as seguintes etapas:

1. Projete a organização das regras e dos blocos de regras no conjunto de regras.

2. Projete o nível em que as diferentes variáveis devem se definir, por exemplo, no nível do conjunto de regras, do bloco de regra ou da regra.
3. Projete os elementos de cada regra com base no padrão de regra.
Esta etapa utiliza os constructos de cada padrão de regra identificado no estágio de planejamento.
4. Projete as interações entre as regras, especialmente em relação ao redirecionamento de eventos e à omissão do processamento do conjunto de regras para eventos em duplicata.
Para obter detalhes adicionais, consulte “Fluxo de Eventos por meio de um Conjunto de Regras” na página 33.
5. Projete, desenvolva e teste quaisquer métodos Java e fragmentos associados criados para serem chamados dentro das expressões.

Continue com “Introdução ao Construtor de Regras”.

Introdução ao Construtor de Regras

Depois de projetar as regras para correlacionar eventos, é possível utilizar o construtor de regras da Tecnologia ACT para gravar as regras.

A seguir, as etapas primárias na gravação de regras com o construtor de regras.

1. Abra o Eclipse Workbench.
2. Configure sua perspectiva no Eclipse Workbench.
3. Configure as suas preferências para a Tecnologia ACT ou aceite as preferências padrão.
4. Crie um projeto para armazenar o arquivo de conjunto de regras ou utilize um projeto existente.
5. Crie um arquivo de conjunto de regras e armazene-o no projeto de sua escolha:
6. Crie pelo menos um bloco de regra no conjunto de regras. Além disso, é possível criar outros blocos de regras no conjunto de regras e pode criar blocos de regras dentro de blocos de regras.
7. Crie regras dentro de blocos de regras.
8. Valide o conjunto de regras.
9. Compile o conjunto de regras.
10. Atualize o conjunto de regras conforme necessário.

É possível incluir, em expressões dentro das regras, os fragmentos da visualização Fragmentos do Eclipse Workbench.

Configurando a Perspectiva no Eclipse Workbench

Antes de fazer alguma coisa, configure a sua perspectiva no Eclipse Workbench. Este tópico descreve como abrir a perspectiva Construtor de Regras.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto.

Embora seja possível utilizar outra perspectiva em vez da Construtor de Regras, as etapas para executar muitas tarefas variarão um pouco, dependendo da perspectiva escolhida.

Para abrir a perspectiva Construtor de Regras, execute as etapas a seguir:

1. No Workbench, clique em **Janela** → **Abrir Perspectiva** → **Outra...**

2. Clique em **Construtor de Regras** e em **OK**. A perspectiva Construtor de Regras é exibida.

Continue com “Configurando as Preferências”.

Configurando as Preferências

Antes de criar um arquivo de conjunto de regras, verifique se as preferências da Tecnologia ACT estão corretamente configuradas no Eclipse Workbench. Este tópico descreve como configurá-las.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto.

Para configurar suas preferências para a Tecnologia ACT, execute as seguintes etapas:

1. No Workbench, clique em **Janela → Preferências...**
2. Clique em **Tecnologia ACT** e, na página Tecnologia ACT, especifique se deseja incluir “act” como um prefixo nas regras e blocos de regras recém-criados no construtor de regras. Por padrão, “act” não é incluído como prefixo.
3. Expanda **Tecnologia ACT**. Dependendo do seu aplicativo, os seguintes itens adicionais podem ser exibidos. Clique em tais itens para configurar as preferências associadas.

Item	Preferência Associada
Provedor de Definição de Evento Base Comum	É possível especificar os arquivos XML que fornecem a estrutura de um ou mais tipos de eventos (incluindo nomes e tipos de atributos que um determinado tipo de evento pode conter) compatíveis à especificação Evento Base Comum. Esses tipos de eventos e atributos estão disponíveis quando as regras são criadas.
Compilador	<p>É possível especificar os seguintes itens primários:</p> <ul style="list-style-type: none">• Se os conjuntos de regras compilados devem ser salvos• O tipo de arquivo dos conjuntos de regras compilados• O caminho de classe do código utilizado na hora da compilação <p>Por padrão, os conjuntos de regras compilados são salvos e mostrados na visualização Navigator com um tipo de arquivo acts, o que indica que a saída do compilador é um conjunto de regras serializado.</p>

Continue com “Criando um Projeto para Armazenar um Arquivo de Conjunto de Regras”.

Criando um Projeto para Armazenar um Arquivo de Conjunto de Regras

No Eclipse Workbench, ao criar um arquivo de conjunto de regras, é necessário especificar o projeto no qual armazenar o arquivo. Portanto, deverá criar um

projeto antes de criar um arquivo de conjunto de regras ou utilizar um projeto existente. Este tópico descreve uma modo de criar um projeto no Eclipse Workbench.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Para criar um projeto simples no Workbench, execute as seguintes etapas:

1. Clique em **Arquivo** → **Novo** → **Projeto....**
2. No assistente para Novo Projeto, clique em **Simple**s → **Projeto** e em **Avançar**.
3. No campo **Nome do Projeto**, digite um nome exclusivo para o projeto. Além disso, utilize o local padrão do projeto ou escolha um local diferente.
4. Clique em **Concluir**. O novo projeto será mostrado na visualização do Navigator da perspectiva Construtor de Regras.

Continue com “Criando um Conjunto de Regras”.

Criando um Conjunto de Regras

Este tópico descreve como criar um conjunto de regras.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Para criar um arquivo de conjunto de regras, execute as seguintes etapas:

1. Clique em **Arquivo** → **Novo** → **Arquivo de Conjunto de Regras**.
2. No assistente Novo Arquivo de Conjunto de Regras, clique no nome da pasta que está associada ao projeto no qual você deseja armazenar o arquivo de conjunto de regras. Provavelmente, esse é o projeto criado em “Criando um Projeto para Armazenar um Arquivo de Conjunto de Regras” na página 39. O nome da pasta é então mostrado no primeiro campo.
3. No campo **Nome do Arquivo**, digite um nome para o arquivo de conjunto de regras. O tipo de arquivo deve ser `actl`.
4. Clique em **Avançar**.
5. Digite um nome exclusivo para o conjunto de regras e uma descrição opcional. Se não quiser utilizar o valor padrão para o campo **Codificação XML**, especifique também o estilo de codificação para o arquivo de conjunto de regras, que é um arquivo XML.
6. Clique em **Concluir**. O arquivo de conjunto de regras é mostrado na pasta `project` na visualização do Navigator. Como o arquivo é validado ao ser salvo, a palavra “Validado” é mostrado ao lado do arquivo. O arquivo do conjunto de regras também é mostrado na visualização Tópicos.

Continue com “Criando um Bloco de Regra”.

Criando um Bloco de Regra

Este tópico descreve como criar um bloco de regra em um conjunto de regras ou em outro bloco de regra.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Se você estiver criando um conjunto de regras pela primeira vez, deverá criar pelo menos um bloco de regra no conjunto de regras antes de criar qualquer regra. Depois de ter criado esse primeiro bloco de regra, poderá criar blocos de regras adicionais, incluindo blocos de regras dentro de blocos de regras.

Para criar um bloco de regra, execute as seguintes etapas:

1. Na visualização Navigator, dê um clique duplo no nome do arquivo do conjunto de regras a ser atualizado. O arquivo será aberto na visualização de Tópicos. Ao clicar no conjunto de regras na visualização de Tópicos, suas informações atualizadas são mostradas na área do editor.
2. Na visualização de Tópicos, clique com o botão direito do mouse no conjunto de regras.
3. Clique em **Novo Filho** → **Bloco de Regra**. Um bloco de regra é então mostrado no conjunto de regras na visualização de Tópicos e as informações atuais para o bloco de regra são mostradas na área do editor.

Agora, é possível criar blocos de regras adicionais das seguintes maneiras:

- Para criar um bloco de regra que é um período para um bloco de regra existente, clique com o botão direito do mouse no bloco de regra existente e clique em **Novo Irmão** → **Bloco de Regra**.
- Para criar um bloco de regra dentro de um bloco existente, clique com o botão direito do mouse no bloco de regra existente e em **Novo Filho** → **Bloco de Regra**.

Também utilizando o editor, é possível atualizar as informações do conjunto de regras e cada bloco de regra. Continue com “Criando uma Regra”.

Criando uma Regra

Este tópico descreve como criar uma regra.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Uma regra deve ser criada em um bloco de regra. Para criar uma regra, execute as seguintes etapas:

1. Na visualização de Tópicos, clique com o botão direito do mouse no bloco de regra que deseja atualizar.
2. Clique em **Novo Filho** e no tipo de regra que deseja criar. A regra é então mostrada no bloco de regra na visualização de Tópicos e as informações atuais para a regra são mostradas na área do editor.

É possível incluir regras adicionais nos blocos de regras da mesma maneira. Também utilizando o editor, é possível atualizar as informações de cada regra. Continue com “Validando um Conjunto de Regras”.

Validando um Conjunto de Regras

Este tópico descreve como validar um conjunto de regras antes de compilá-lo.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Para validar um conjunto de regras, execute as seguintes etapas:

1. Na visualização Tópicos, clique com o botão direito do mouse numa regra, bloco de regra ou conjunto de regras.

2. Clique em **Validar Conjunto de Regras**. Quando a validação estiver concluída, uma janela de mensagens será mostrada para indicar se existem problemas. Se a validação for concluída com êxito, a palavra “Validada” será mostrada à direita do nome do arquivo do conjunto de regras na visualização Navigator.

Quando a validação concluir com êxito, prossiga para “Compilando um Conjunto de Regras”.

Compilando um Conjunto de Regras

Este tópico descreve como compilar um conjunto de regras.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Na visualização Navigator ou Tópicos, clique com o botão direito do mouse no conjunto de regras a ser compilado e selecione **Compilar Conjunto de Regras**. Os erros de compilação são mostrados na visualização Problemas.

Por padrão, se não houver nenhum erro de compilação, a regra compilada será mostrada na visualização Navigator com um tipo de arquivo padrão, `acts`, que indica um conjunto de regras serializado. Além disso, na visualização Navigator, a palavra “Compilado” será mostrada à direita do nome do arquivo do conjunto de regras.

Atualizando um Conjunto de Regras

Este tópico descreve como atualizar um conjunto de regras.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

Na visualização de Tópicos, clique no item (regra, bloco de regra ou conjunto de regras) que deseja atualizar. As informações atuais para o item serão então mostradas na área do editor e você poderá utilizar o editor para atualizá-las.

Para criar um bloco de regra ou uma regra que seja um período para um bloco de regra ou uma regra existente, execute as seguintes etapas:

1. Clique com o botão direito do mouse no bloco de regra ou na regra existente.
2. Clique em **Novo Irmão** e no item (bloco de regra ou tipo de regra) que deseja incluir.

Para criar um bloco de regra ou uma regra em um bloco de regra existente, execute as seguintes etapas:

1. Clique com o botão direito do mouse no bloco de regra existente.
2. Clique em **Novo Filho** e no item (bloco de regra ou tipo de regra) que deseja incluir.

Para acessar outras funções de atualização na visualização de Tópicos, execute as seguintes etapas:

1. Clique com o botão direito do mouse no item (bloco de regra ou regra) que deseja atualizar.
2. Clique no item de menu da função, como **Excluir**, **Copiar** ou **Colar**.

Incluindo Fragmentos em Expressões Dentro das Regras

É possível incluir, em expressões dentro das regras, os fragmentos da visualização Fragmentos do Eclipse Workbench.

Primeiramente, abra o Eclipse Workbench, se ainda não estiver aberto. Abra, também, a perspectiva Construtor de Regras.

A visualização Fragmentos é mostrada na perspectiva Construtor de Regras. Os fragmentos estão organizados em categorias com base na sua função.

Para incluir um fragmento a partir da visualização Fragmentos, execute as seguintes etapas:

1. Clique numa categoria de fragmento para ver os nomes dos fragmentos que ela contém.
2. Clique no fragmento a ser incluído numa expressão.
3. Arraste-o para o respectivo campo **Expression**. O código é inserido na posição do cursor no campo **Expression**. Se o código exigir uma inserção do autor da regra, por exemplo, o texto da mensagem ou os valores das variáveis, você receberá um aviso para fazê-la antes de o código ser incluído na expressão.

Continue com “Validando um Conjunto de Regras” na página 41.

Parte 2. Referência do Autor de Regras

Capítulo 4. Resumo da Organização do Conjunto de Regras

Esta referência lista todos os elementos da linguagem de um conjunto de regras, um bloco de regra e cada tipo de regra. Ela serve como referência rápida para codificação de um conjunto de regras.

Tabela 10 explica o significado da notação que segue cada elemento de linguagem. n representa um número ilimitado.

Tabela 10. Explicação da Notação que Define o Número de Ocorrências de um Elemento de Linguagem

Notação	Significado
(0, 1)	0 indica que o elemento de linguagem é opcional. 1 indica que, se codificado, apenas 1 ocorrência será permitida.
(0, n)	0 indica que o elemento de linguagem é opcional. n indica que, se codificado, várias ocorrências serão permitidas.
(1, 1)	O primeiro 1 indica que o elemento de linguagem é obrigatório. O segundo 1 indica que apenas 1 ocorrência é permitida.
(1, n)	1 indica que o elemento de linguagem é obrigatório. n indica que várias ocorrências são permitidas.
(2, n)	2 indica que 2 ocorrências do elemento de linguagem são obrigatórias. n indica que ocorrências adicionais são permitidas.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Resumo do Conjunto de Regras

Este resumo lista os elementos de linguagem de um conjunto de regras.

Elementos do Conjunto de Regras

<ruleSet> contém os seguintes elementos:

- <comment> (0, 1)
- <import> (0, n)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <ruleBlock> (0, n)

Referências relacionadas

“Resumo do Bloco de Regra”

Este resumo lista os elementos de linguagem de um bloco de regra.

Resumo do Bloco de Regra

Este resumo lista os elementos de linguagem de um bloco de regra.

Elementos do bloco de regra

<ruleBlock> contém os elementos a seguir.

Se estiverem codificados, os elementos <comment>, <import> e <variable> deverão ser codificados na ordem mostrada. Os outros elementos poderão ser codificados em qualquer ordem.

- <comment> (0, 1)
- <import> (0, n)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <ruleBlock> (0, n)
- <collectionRule> (0, n)
- <computationRule> (0, n)
- <duplicateRule> (0, n)
- <filterRule> (0, n)
- <sequenceRule> (0, n)
- <thresholdRule> (0, n)
- <timerRule> (0, n)

Referências relacionadas

“Resumo da Regra de Coleta”

Este resumo lista todos os elementos da linguagem da regra de coleta.

“Resumo da Regra de Computação” na página 50

Este resumo lista todos os elementos da linguagem da regra de computação.

“Resumo da Regra de Duplicata” na página 51

Este resumo lista todos os elementos da linguagem da regra de duplicação.

“Resumo da Regra de Filtragem” na página 52

Este resumo lista todos os elementos da linguagem da regra de filtragem.

“Resumo da Regra de Sequência” na página 53

Este resumo lista todos os elementos da linguagem da regra de sequência.

“Resumo da Regra de Limite” na página 55

Este resumo lista todos os elementos da linguagem da regra de limite.

“Resumo da Regra de Cronômetro” na página 56

Este resumo lista todos os elementos da linguagem da regra de cronômetro.

Resumo da Regra de Coleta

Este resumo lista todos os elementos da linguagem da regra de coleta.

Elementos de Regra

<collectionRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)

- <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
- <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
- <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <activationByGroupingKey> (0, 1)
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <stopAfter> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
- <eventSelector> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <groupingKey> (0, 1)
 - Os três seguintes elementos, em qualquer ordem (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>

- <runUntilDeactivated>
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Resumo da Regra de Computação

Este resumo lista todos os elementos da linguagem da regra de computação.

Elementos de Regra

<computationRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <activationByGroupingKey> (0, 1)
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <stopAfter> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)

- <onDeactivation> (0, 1)
 - <action> (0, n)
- <onUnload> (0, 1)
 - <action> (0, n)
- <eventSelector> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <groupingKey> (0, 1)
 - Os três seguintes elementos, em qualquer ordem (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <computeFunction> (1, 1)
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Resumo da Regra de Duplicata

Este resumo lista todos os elementos da linguagem da regra de duplicação.

Elementos de Regra

<duplicateRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)

- <filteringPredicate> (0, 1)
- <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <activationByGroupingKey> (0, 1)
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <stopAfter> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
- <eventSelector> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <groupingKey> (0, 1)
 - Os três seguintes elementos, em qualquer ordem (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onNextEvent> (0, 1)
 - <action> (0, n)
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Resumo da Regra de Filtragem

Este resumo lista todos os elementos da linguagem da regra de filtragem.

Elementos de Regra

<filterRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
 - <eventSelector> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <onDetection> (0, 1)
 - <action> (0, n)

Resumo da Regra de Seqüência

Este resumo lista todos os elementos da linguagem da regra de seqüência.

Elementos de Regra

<sequenceRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <activationByGroupingKey> (0, 1)
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <stopAfter> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
 - <eventSelector> (2, n)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)

- <groupingKey> (0, 1)
 - Os três seguintes elementos, em qualquer ordem (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onTimeOut> (0, 1)
 - <action> (0, n)

Resumo da Regra de Limite

Este resumo lista todos os elementos da linguagem da regra de limite.

Elementos de Regra

<thresholdRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <activationByGroupingKey> (0, 1)
 - <activateOnEvent> (0, 1)

- <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <stopAfter> (0, 1)
- <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
- <eventSelector> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <groupingKey> (0, 1)
 - Os três seguintes elementos, em qualquer ordem (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- Um dos três seguintes elementos (1, 1):
 - <booleanThreshold>
 - <computedThreshold>
 - <eventCountThreshold>
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onTimeOut> (0, 1)
 - <action> (0, n)

Resumo da Regra de Cronômetro

Este resumo lista todos os elementos da linguagem da regra de cronômetro.

Elementos de Regra

<timerRule> contém os seguintes elementos:

- <comment> (0, 1)
- <variable> (0, n)

- <comment> (0, 1)
- <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Um dos três seguintes elementos (1, 1):
 - <dateTime>
 - <never>
 - <after>
 - <activateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
 - <deactivateOnEvent> (0, 1)
 - <eventType> (0, n)
 - <filteringPredicate> (0, 1)
- <lifeCycleActions> (0, 1)
 - <onLoad> (0, 1)
 - <action> (0, n)
 - <onActivation> (0, 1)
 - <action> (0, n)
 - <onDeactivation> (0, 1)
 - <action> (0, n)
 - <onUnload> (0, 1)
 - <action> (0, n)
- <timeWindow> (1, 1)
 - Um dos dois seguintes elementos (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Capítulo 5. Referência de Elemento de Linguagem

Essa referência descreve os detalhes dos elementos de linguagem no esquema XML da linguagem de regras da Tecnologia ACT. Os elementos de linguagem são listados em ordem alfabética e os atributos disponíveis para cada elemento são descritos no tópico desse elemento.

Em XML e em outras linguagens de marcações, como SGML e HTML, um elemento é uma unidade básica que consiste em uma tag de início, uma tag de encerramento, atributos associados e seus valores e qualquer texto contido entre as duas. Um atributo é um par nome-valor codificado em um elemento para definir uma determinada característica do elemento. Um atributo possui um tipo de dado que identifica o tipo de informações fornecidas em seu valor (por exemplo, informações numéricas, textuais ou booleanas).

Em XML, um espaço de nomes é um URI (Uniform Resource Identifier) que fornece um nome exclusivo para associar aos elementos e definições de tipos em um esquema. O URI indica qual esquema XML contém a definição de um elemento. Um espaço de nomes é especificado com uma cadeia de prefixo seguida por dois pontos. O esquema da linguagem de regras da Tecnologia ACT é definido em três arquivos diferentes e utiliza os seguintes três espaços de nomes:

- xsd:** Esse espaço de nomes indica que o elemento de linguagem é definido no esquema XML padrão, descrito em <http://www.w3.org>.
- br:** Esse espaço de nomes indica que o elemento de linguagem é definido no esquema de conjunto de regras base da Tecnologia ACT, que está no arquivo ACTparser.jar, no subdiretório com/ibm/correlation/ruleparser/xml/RuleSetBase.xsd. Por exemplo, `br:ruleSet` refere-se ao elemento `ruleSet` definido no arquivo `RuleSetBase.xsd`.
- act:** Esse espaço de nomes indica que o elemento de linguagem é definido no esquema de linguagem da Tecnologia ACT, que está no arquivo ACTparser.jar, no subdiretório com/ibm/correlation/ruleparser/xml/ACTL.xsd. Por exemplo, `act:ruleSet` refere-se ao elemento `ruleSet` definido no arquivo `ACTL.xsd`.

No esquema da linguagem de regras, os elementos de linguagem são definidos como elementos ou tipos complexos, por exemplo:

```
<xsd:element name="symbol" minOccurs="1" maxOccurs="unbounded"></element>
<xsd:complexType name="symbol"></complexType>
```

No esquema, os atributos `minOccurs` e `maxOccurs` definem os números mínimo e máximo de ocorrências, respectivamente, para um elemento de linguagem. A Tabela 11 descreve o significado de valores diferentes dos atributos `minOccurs` e `maxOccurs`.

Tabela 11. Atributos no Esquema que Definem o Número de Ocorrências de um Elemento de Linguagem

Atributo	Valor do atributo	Significado
<code>minOccurs</code>	0	O elemento de linguagem é opcional.

Tabela 11. Atributos no Esquema que Definem o Número de Ocorrências de um Elemento de Linguagem (continuação)

Atributo	Valor do atributo	Significado
minOccurs	1	O elemento de linguagem deve ocorrer pelo menos uma vez. 1 é o valor padrão do atributo minOccurs.
minOccurs	2	O elemento de linguagem deve ocorrer pelo menos duas vezes.
maxOccurs	1	O elemento de linguagem não pode ocorrer mais de uma vez. 1 é o valor padrão do atributo maxOccurs.
maxOccurs	ilimitado	O elemento de linguagem pode ocorrer qualquer número de vezes.

Elemento action

O elemento <action> contém uma expressão que define uma ação de resposta da regra ou uma ação de ciclo de vida.

Detalhes

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

<action> possui os seguintes atributos:

Tabela 12. Atributos do Elemento <action>

Nome	Descrição	Tipo de dado	Obrigatório?
expressionLanguage	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é java.	xsd:NMTOKEN	Sim
name	Identifica a ação. Esse identificador pode ser útil para resolução de problemas, principalmente se for um nome exclusivo entre todos os elementos <action> definidos para uma ação específica de resposta de regra ou de ciclo de vida.	xsd:NMTOKEN	Não

Contido em

<action> está contido nos seguintes elementos:

- <onActivation>
- <onDeactivation>
- <onDetection>
- <onLoad>
- <onNextEvent>

- <onTimeOut>
- <onTimeWindowComplete>
- <onUnload>

Contém

<action> não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento activateOnEvent

O elemento <activateOnEvent> define os eventos que podem ativar a regra ou, para regras definidas com um elemento <groupingKey>, uma instância de regra.

A seguir, as três maneiras possíveis de seleção de eventos:

- A utilização de um ou mais elementos <eventType> com um elemento <filteringPredicate>
- A utilização de um ou mais elementos <eventType> sem um elemento <filteringPredicate>
- A utilização de um elemento <filteringPredicate> sem nenhum elemento <eventType>

Se a regra estiver inativa e nenhum elemento <eventType> ou <filteringPredicate> estiver codificado, qualquer evento que ocorrer será selecionado.

A não codificação de nenhum elemento <eventType> pode afetar negativamente o desempenho do sistema.

Suponha que você queira selecionar todos os eventos do tipo Falha na Auditoria. É possível utilizar um predicado de filtragem para aperfeiçoar ainda mais os critérios de seleção para incluir apenas os eventos que possuam um atributo de evento com um determinado valor. Por exemplo, você codificaria um elemento <eventType> para selecionar todos os eventos do tipo Falha na Auditoria e codificaria um elemento <filteringPredicate> para selecionar apenas os eventos que possuem um atributo de nome do host com o valor MyCriticalSystem.

Atributos

<activateOnEvent> não possui atributos.

Contido em

<activateOnEvent> está contido nos seguintes elementos:

- <activationInterval>
- <activationByGroupingKey>

Contém

<activateOnEvent> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 13. Elementos Contidos no Elemento <activateOnEvent>

Elemento	Obrigatório ou opcional?
<eventType>	Opcional. 0 ou mais ocorrências são permitidas.
<filteringPredicate>	Opcional. 0 ou 1 ocorrência é permitida.
<stopAfter>	Esse elemento é válido apenas quando o elemento <activateOnEvent> está contido no elemento <activationByGroupingKey>. Opcional. 0 ou 1 ocorrência é permitida.

Elemento activationByGroupingKey

O elemento <activationByGroupingKey> contém elementos que especificam os eventos que podem ativar e desativar uma instância de regra definida pelo elemento <groupingKey>. Como o elemento <groupingKey> não é válido para as regras de filtragem e de cronômetro, o elemento <activationByGroupingKey> não se aplica a essas regras.

Detalhes

A função fornecida pelo elemento <activationByGroupingKey> é para ser utilizada em regras na qual você define uma chave de agrupamento. Ela permite controlar a ativação e a desativação de instâncias de regras com base na chave de agrupamento. Quando você codifica o elemento <activationByGroupingKey>, cada instância da regra pode ser ativada e desativada individualmente com base nas condições <activateOnEvent> e <deactivateOnEvent> no <activationByGroupingKey>.

O exemplo a seguir ilustra a utilização do elemento <activationByGroupingKey> em uma regra de computação.

- A regra de computação a seguir aceita eventos do tipo StockSharesTraded. Esses eventos indicam o número de ações de capital que estão sendo comercializadas para várias empresas diferentes.
- A chave de agrupamento é o atributo stockSymbol de um evento. O valor do atributo stockSymbol é o nome de uma empresa específica.
- O valor do atributo sharesTraded de um evento é o número de ações de capital comercializadas para a respectiva empresa (aquela indicada pelo valor do atributo stockSymbol).
- Para uma empresa específica, é criado um relatório indicando o número de ações de capital que foram comercializadas para ela durante um período de 10 minutos. No entanto, antes da regra de cálculo poder criar esse relatório, ela deverá receber um evento do tipo StartReporting, com o nome da respectiva empresa como o valor do atributo stockSymbol.

```
<computationRule name="StockReporter">  
  <variable dataType="java.lang.Integer" name="totalSharesTraded">  
    <varInitializer expressionLanguage="java">
```

```

        return new Integer(0);
    </varInitializer>
</variable>

<activationInterval>
  <activationTime>
    <start>
      <inactiveWhenLoaded/>
    </start>
  </activationTime>
  <activationByGroupingKey>
    <activateOnEvent>
      <eventType type="StartReporting"/>
    </activateOnEvent>
  </activationByGroupingKey>
</activationInterval>

<eventSelector>
  <eventType type="StockSharesTraded"/>
</eventSelector>

<groupingKey>
  <attributeName>stockSymbol</attributeName>
</groupingKey>

<computeFunction assignTo="totalSharesTraded" expressionLanguage="java">
  return new Integer(act_lib.getIntVariable("totalSharesTraded")
    + act_event.getIntAttribute("sharesTraded"));
</computeFunction>

<timeWindow>
  <timeInterval unit="ISO-8601" duration="PT10M"/>
</timeWindow>

<onTimeWindowComplete>
  <action expressionLanguage="java">
    StockReport.createReport(act_eventList.get(0).getStringAttribute("stockSymbol"),
      act_lib.getIntVariable("totalSharesTraded"));
  </action>
</onTimeWindowComplete>
</computationRule>

```

Atributos

<activationByGroupingKey> não possui atributos.

Contido em

<activationByGroupingKey> está contido no seguinte elemento:

- <activationInterval>

Contém

<activationByGroupingKey> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 14. Elementos Contidos no Elemento <activationByGroupingKey>

Elemento	Obrigatório ou opcional?
<activateOnEvent>	Opcional. Ocorrência 0 ou 1 é permitida.

Tabela 14. Elementos Contidos no Elemento <activationByGroupingKey> (continuação)

Elemento	Obrigatório ou opcional?
<deactivateOnEvent>	Opcional. Ocorrência 0 ou 1 é permitida.

Relacionamento entre os Elementos <activationInterval> e <activationByGroupingKey>

Os elementos <activateOnEvent> e <deactivateOnEvent> estão contidos nestes dois elementos:

- <activationInterval>
- <activationByGroupingKey>, que também está contido no <activationInterval>

O comportamento da regra pode ser diferente com base na atividade da regra atual e nas interações entre as definições de <activateOnEvent> e <deactivateOnEvent> nos elementos <activationInterval> e <activationByGroupingKey>. O exemplo a seguir ilustra como essas definições podem interagir.

Nesse exemplo, foi definida uma regra de duplicata para remover os eventos de sistemas que estão no modo de manutenção e para fornecer, no final do período de manutenção, um relatório de resumo com o número de eventos removidos.

Por padrão, uma regra na qual uma chave de agrupamento está definida permite que todos os valores da chave de agrupamento sejam processados. Portanto, quando os eventos atendem aos critérios de seleção de evento da regra, todas as instâncias de regras ficam ativas e prontas para aceitar esses eventos de acordo com qualquer valor da chave de agrupamento. O intervalo de ativação da regra é o mesmo que seria se a regra não tivesse uma chave de agrupamento porque, em essência, todos os eventos que atenderem aos critérios de seleção de eventos da regra serão processados.

No exemplo a seguir, a chave de agrupamento é hostname e as definições no elemento <activationInterval> especificam as seguintes ações:

1. Ativam todas as instâncias de regras quando um evento do tipo StartMaintenanceModeAllHosts é recebido.
2. Desativam todas as instâncias de regras depois de 2 horas ou quando um evento do tipo StopMaintenanceModeAllHosts é recebido.

```
<duplicateRule name="Maintenance_Supression">
  <activationInterval>
    <activationTime>
      <start>
        <inactiveWhenLoaded/>
      </start>
      <stop>
        <after duration="PT2H" unit="ISO-8601"/>
      </stop>
    </activationTime>
    <activateOnEvent>
      <eventType type="StartMaintenanceModeAllHosts"/>
    </activateOnEvent>
    <deactivateOnEvent>
      <eventType type="StopMaintenanceModeAllHosts"/>
    </deactivateOnEvent>
  </activationInterval>
  <groupingKey missingAttributeHandling="ignoreEvent">
    <attributeName>hostname</attributeName>
  </groupingKey>
  <timeWindow>
    <runUntilDeactivated/>
  </timeWindow>
  <onDetection>
```



```

<action expressionLanguage="java" name="DropEvent">
  <![CDATA[act_lib.exitRuleSet();]]>
</action>
</onDetection>
<onTimeWindowComplete>
  <action expressionLanguage="java" name="CreateSummaryOfSupressedEvents">
    <![CDATA[Helper.createSummaryEvent("MaintenanceSummary", act_eventList, act_lib);]]>
  </action>
</onTimeWindowComplete>
</duplicateRule>

```

Em algumas situações, é possível que você queira controlar quais instâncias de regras ficarão ativas e quando ficarão ativas. Nesses casos, é necessário codificar o elemento `<activationByGroupingKey>`.

O exemplo a seguir estende o exemplo anterior e ilustra como é possível utilizar o valor da chave de agrupamento para selecionar quais instâncias de regras podem ser processadas. As definições no elemento `<activationByGroupingKey>` especificam as seguintes ações:

1. Permitem que as instâncias de regras para nomes específicos do host sejam processadas quando eventos do tipo `StartMaintenanceMode` são recebidos para esses nomes de host.
2. Desativam essas instâncias de regras depois de 2 horas de ativação ou quando um evento do tipo `StopMaintenanceMode` é recebido para o respectivo nome do host.

```

<activationByGroupingKey>
  <activateOnEvent>
    <eventType type="StartMaintenanceMode"/>
    <stopAfter duration="PT2H" unit="ISO-8601"/>
  </activateOnEvent>
  <deactivateOnEvent>
    <eventType type="StopMaintenanceMode"/>
  </deactivateOnEvent>
</activationByGroupingKey>

```

As instruções a seguir resumem o que acontece quando você codifica o elemento `<activationByGroupingKey>`:

- Quando o elemento `<activateOnEvent>` é codificado no elemento `<activationByGroupingKey>`, apenas os eventos que compartilham o valor da chave de agrupamento do evento que atende à condição `<activationByGroupingKey>` `<activateOnEvent>` podem ser processados.
- Quando o elemento `<deactivateOnEvent>` é codificado no elemento `<activationByGroupingKey>`, os eventos que compartilham o valor da chave de agrupamento do evento que atende à condição `<activationByGroupingKey>` `<deactivateOnEvent>` *não* podem ser processados.

O Efeito de Diferentes Definições de Ativação e Desativação no Estado da Regra

Tabela 15 na página 66 e Tabela 16 na página 68 mostram como o estado de uma regra é afetado por diferentes definições de ativação e desativação. Nessas tabelas, as seguintes convenções são utilizadas:

- *A* é um evento de ativação.
- Na notação "*A*[*x*]", *x* representa o valor da chave de agrupamento. Por exemplo, *A*[1] é um evento de ativação com uma chave de agrupamento de valor 1.
- *D* é o evento de desativação.

- Na notação “ $D[x]$ ”, x representa o valor da chave de agrupamento. Por exemplo, $D[1]$ é um evento de desativação com uma chave de agrupamento de valor 1.

Tabela 15. Alterações do Estado da Regra com Base em Diferentes Definições de Ativação

Estado inicial da regra	O estado da regra é afetado potencialmente pelo	Estado final da regra
Inativo	Tempo definido em <code><activationInterval></code> <code><activationTime></code> <code><start></code>	<ol style="list-style-type: none"> 1. A regra é ativada. 2. As ações de <code><onActivation></code> são executadas. 3. Todos os valores da chave de agrupamento são permitidos.
	Método <code>activate()</code>	
	Evento A , definido em <code><activationInterval></code> <code><activateOnEvent></code>	
	Evento $A[1]$, definido em <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>sem</i> <code><stopAfter></code>)	<ol style="list-style-type: none"> 1. A regra é ativada. 2. As ações de <code><onActivation></code> são executadas. 3. Apenas a chave de agrupamento de valor 1 é permitida. Quando o padrão de regra é correspondido para essa instância de regra, a chave de agrupamento de valor 1 não é mais permitida.
<ul style="list-style-type: none"> • Ativo • Permite todos os valores da chave de agrupamento 	Tempo definido em <code><activationInterval></code> <code><activationTime></code> <code><start></code>	<p>Não ocorreu nenhuma alteração no estado da regra. É o mesmo estado da regra de início.</p>
	Método <code>activate()</code>	
	Evento A , definido em <code><activationInterval></code> <code><activateOnEvent></code>	
	Evento $A[1]$, definido em <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>sem</i> <code><stopAfter></code>)	
	Evento $A[2]$, definido em <code><activateOnEvent></code> (<i>com</i> <code><stopAfter></code>)	

Tabela 15. Alterações do Estado da Regra com Base em Diferentes Definições de Ativação (continuação)

Estado inicial da regra	O estado da regra é afetado potencialmente pelo	Estado final da regra
<ul style="list-style-type: none"> • Ativo • Permite apenas valores da chave de agrupamento que ativaram instâncias de regras com base nas definições <code><activationByGroupingKey></code> <code><activateOnEvent></code> 	Tempo definido em <code><activationInterval></code> <code><activationTime></code> <code><start></code>	Não ocorreu nenhuma alteração no estado da regra. É o mesmo estado da regra de início.
	Método <code>activate()</code>	
	Evento <i>A</i> , definido em <code><activationInterval></code> <code><activateOnEvent></code>	Todos os valores da chave de agrupamento são permitidos.
	Evento <i>A</i> [1], definido em <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>sem</i> <code><stopAfter></code>)	<ul style="list-style-type: none"> • Agora a chave de agrupamento de valor 1 é permitida, além dos valores permitidos anteriormente. • Quando o padrão de regra é correspondido para essa instância de regra, a chave de agrupamento de valor 1 não é mais permitida.
<ul style="list-style-type: none"> • Ativo • Permite todos os valores da chave de agrupamento, exceto aqueles não permitidos com base nas definições <code><activationByGroupingKey></code> <code><deactivateOnEvent></code> 	Tempo definido em <code><activationInterval></code> <code><activationTime></code> <code><start></code>	Não ocorreu nenhuma alteração no estado da regra. É o mesmo estado da regra de início.
	Método <code>activate()</code>	
	Evento <i>A</i> , definido em <code><activationInterval></code> <code><activateOnEvent></code>	Todos os valores da chave de agrupamento são permitidos.
	Evento <i>A</i> [1], definido em <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>sem</i> <code><stopAfter></code>)	Agora a chave de agrupamento de valor 1 é permitida, além dos valores permitidos anteriormente.
	Evento <i>A</i> [2], definido em <code><activateOnEvent></code> (<i>com</i> <code><stopAfter></code>)	<ul style="list-style-type: none"> • Agora a chave de agrupamento de valor 2 é permitida, além dos valores permitidos anteriormente. • Esse valor será permitido apenas durante o valor especificado pelo elemento <code><stopAfter></code>. • O padrão de regra para essa instância de regra pode ser correspondido várias vezes nessa duração.
	Evento <i>A</i> [2], definido em <code><activateOnEvent></code> (<i>com</i> <code><stopAfter></code>)	Agora a chave de agrupamento de valor 2 é permitida, além dos valores permitidos anteriormente.

Tabela 16. Alterações do Estado da Regra com Base em Diferentes Definições de Desativação

Estado inicial da regra	O estado da regra é afetado potencialmente pelo	Estado final da regra
Inativo	Tempo definido em <activationInterval> <activationTime> <stop>	Não ocorreu nenhuma alteração no estado da regra. É o mesmo estado da regra de início.
	Método deactivate()	
	Evento <i>D</i> , definido em <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definido em <activationByGroupingKey> <deactivateOnEvent>	
	Tempo de duração definido em <activationByGroupingKey> <activateOnEvent> <stopAfter> termina para o evento <i>A</i> [2]	
<ul style="list-style-type: none"> Ativo Permite todos os valores da chave de agrupamento 	Tempo definido em <activationInterval> <activationTime> <stop>	<ol style="list-style-type: none"> Todas as instâncias de regras são desativadas. As ações de <onDeactivation> são executadas. A regra é desativada.
	Método deactivate()	
	Evento <i>D</i> , definido em <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definido em <activationByGroupingKey> <deactivateOnEvent>	<ul style="list-style-type: none"> A chave de agrupamento de valor 1 não é mais permitida. Se a instância de regra com uma chave de agrupamento de valor 1 estiver ativa, ela será desativada.
	Tempo de duração definido em <activationByGroupingKey> <activateOnEvent> <stopAfter> termina para o evento <i>A</i> [2]	A instância de regra com uma chave de agrupamento de valor 2 é desativada.
<ul style="list-style-type: none"> Ativo Permite apenas valores da chave de agrupamento que ativaram instâncias de regras com base nas definições <activationByGroupingKey> <activateOnEvent> 	Tempo definido em <activationInterval> <activationTime> <stop>	<ol style="list-style-type: none"> Todas as instâncias de regras são desativadas. As ações de <onDeactivation> são executadas. A regra é desativada.
	Método deactivate()	
	Evento <i>D</i> , definido em <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definido em <activationByGroupingKey> <deactivateOnEvent>	<ul style="list-style-type: none"> A chave de agrupamento de valor 1 não é mais permitida. Se a instância de regra com uma chave de agrupamento de valor 1 estiver ativa, ela será desativada.
	Tempo de duração definido em <activationByGroupingKey> <activateOnEvent> <stopAfter> termina para o evento <i>A</i> [2]	<ul style="list-style-type: none"> A chave de agrupamento de valor 2 não é mais permitida. A instância de regra com uma chave de agrupamento de valor 2 é desativada.

Tabela 16. Alterações do Estado da Regra com Base em Diferentes Definições de Desativação (continuação)

Estado inicial da regra	O estado da regra é afetado potencialmente pelo	Estado final da regra
<ul style="list-style-type: none"> Ativo Permite todos os valores da chave de agrupamento, exceto aqueles não permitidos com base nas definições <code><activationByGroupingKey></code> <code><deactivateOnEvent></code> 	Tempo definido em <code><activationInterval></code> <code><activationTime></code> <code><stop></code>	<ol style="list-style-type: none"> Todas as instâncias de regras são desativadas. As ações de <code><onDeactivation></code> são executadas. A regra é desativada.
	Método <code>deactivate()</code>	
	Evento <i>D</i> , definido em <code><activationInterval></code> <code><deactivateOnEvent></code>	
	Evento <i>D</i> [1], definido em <code><activationByGroupingKey></code> <code><deactivateOnEvent></code>	<ul style="list-style-type: none"> A chave de agrupamento de valor 1 não é mais permitida. Se a instância de regra com uma chave de agrupamento de valor 1 estiver ativa, ela será desativada.
	Tempo de duração definido em <code><activationByGroupingKey></code> <code><activateOnEvent></code> <code><stopAfter></code> termina para o evento <i>A</i> [2]	A instância de regra com uma chave de agrupamento de valor 2 é desativada.

Elemento `activationInterval`

O elemento `<activationInterval>` contém elementos que definem quando uma regra está ativa e inativa.

Detalhes

Uma regra pode ser ativada ou desativada em um momento exato distinto ou por um evento específico.

Se você especificar que uma regra deve ser ativada ou desativada em um momento exato distinto e por um evento específico, a regra será ativada ou desativada, o que ocorrer primeiro, o momento exato ou o recebimento do evento. No entanto, nesse caso, a regra poderá ser ativada ou desativada por vários eventos durante seu ciclo de vida. Por exemplo, uma regra pode ser ativada por um evento, desativada, ativada em um momento exato definido, desativada novamente e ativada por outro evento.

Em um ambiente de negócios, talvez você queira ativar uma regra ao receber um evento que indica se a bolsa de valores está aberta para negócios. Em um ambiente de TI, talvez você queira iniciar uma janela de manutenção às 6h em 29 de outubro de 2005 e finalizá-la em um dos seguintes horários, com base no que ocorrer primeiro:

- 11h30 em 30 de outubro de 2005
- Quando um evento é recebido indicando que o trabalho de manutenção foi concluído

Atributos

`<activationInterval>` não possui atributos.

Contido em

<activationInterval> está contido nos seguintes elementos:

- <collectionRule>
- <computationRule>
- <duplicateRule>
- <filterRule>
- <sequenceRule>
- <thresholdRule>
- <timerRule>

Contém

<activationInterval> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 17. Elementos Contidos no Elemento <activationInterval>

Elemento	Obrigatório ou opcional?
<activationTime>	Opcional. Ocorrência 0 ou 1 é permitida.
<activateOnEvent>	Opcional. Ocorrência 0 ou 1 é permitida.
<deactivateOnEvent>	Opcional. Ocorrência 0 ou 1 é permitida.
<activationByGroupingKey>	Opcional. Ocorrência 0 ou 1 é permitida.

Relacionamentos entre os Elementos Contidos

Os elementos <start> e <stop> contidos no elemento <activationTime> são um método estático de ativação e desativação de uma regra. Por meio desses elementos, uma regra é ativada ou desativada em um momento exato distinto. Por outro lado, os elementos <activateOnEvent> e <deactivateOnEvent> são um método dinâmico de ativação e desativação de uma regra. Por meio desses elementos, uma regra será ativada ou desativada se um determinado evento ocorrer. Por exemplo, uma regra será ativada por qualquer evento que atenda aos critérios definidos para o elemento <activateOnEvent>, se a regra ainda não estiver ativa. Uma regra será desativada por qualquer evento que atenda aos critérios definidos para o elemento <deactivateOnEvent>, se a regra ainda não estiver inativa. Portanto, determinados eventos podem alterar a definição estática de quando uma regra será ativada ou desativada.

Tabela 18 na página 71 descreve como e quando uma regra está ativada ou desativada com base em determinadas combinações nas quais os seguintes elementos podem ser codificados:

- <start>
- <stop>
- <activateOnEvent>
- <deactivateOnEvent>

Em Tabela 18 na página 71, X representa o nome de um evento que ativa a regra e Y representa o nome de um evento que desativa a regra.

Se o elemento <start> não estiver mesmo codificado, a hora de início padrão será a mesma definida pelo elemento <whenLoaded>.

Se o elemento <stop> não estiver mesmo codificado, a hora de parada padrão será a mesma definida pelo elemento <never>.

Tabela 18. Atividade da Regra Baseada em Diferentes Combinações de Codificação dos Elementos Contidos no <activationInterval>

<activationTime>		<activateOnEvent>	<deactivateOnEvent>	Atividade da regra
<start>	<stop>			
<whenLoaded>	<never>			A regra está ativa quando é carregada e permanece ativa durante a execução do mecanismo da Tecnologia ACT.
<whenLoaded>	<never>		Y	A regra está ativa quando é carregada. O evento Y desativa a regra.
<whenLoaded>	<never>	X	Y	A regra está ativa quando é carregada. O evento Y desativa a regra e o evento X a reativa. Essa desativação e reativação podem ocorrer várias vezes.
<whenLoaded>	<after>			A regra está ativa quando é carregada e é desativada depois de um intervalo de tempo especificado.
<whenLoaded>	<dateTime>			A regra está ativa quando é carregada e é desativada depois de uma data e hora especificadas.
<inactiveWhenLoaded>	<never>	X		A regra está inativa quando é carregada. O evento X ativa a regra e esta permanece ativa durante a execução do mecanismo da Tecnologia ACT.
<inactiveWhenLoaded>	<never>	X	Y	A regra está inativa quando é carregada. O evento X ativa a regra e o evento Y a desativa. Essa ativação e desativação podem ocorrer várias vezes.
<dateTime>	<dateTime>			A regra é ativada em uma data e hora especificadas e é desativada em uma data e hora especificadas.
<dateTime>	<dateTime>	X	Y	A regra é ativada em uma data e hora especificadas e é desativada em uma data e hora especificadas. O evento X ativa a regra e o evento Y a desativa. Os eventos X e Y podem ativar e desativar a regra várias vezes.
<dateTime>	<never>			A regra é ativada em uma data e hora especificadas e permanece ativa durante a execução do mecanismo da Tecnologia ACT.
<dateTime>	<never>		Y	A regra é ativada em uma data e hora especificadas. O evento Y desativa a regra.
<dateTime>	<never>	X	Y	A regra é ativada em uma data e hora especificadas. O evento Y desativa a regra e o evento X a reativa. Essa desativação e reativação podem ocorrer várias vezes.
<dateTime>	<after>			A regra é ativada em uma data e hora especificadas e é desativada depois de um intervalo de tempo especificado.
<dateTime>	<after>	X	Y	A regra é ativada em uma data e hora especificadas e é desativada depois de um intervalo de tempo especificado. O evento X ativa a regra e o evento Y a desativa. Essa ativação e desativação podem ocorrer várias vezes.

Elemento activationTime

O elemento <activationTime> define os momentos exatos distintos quando uma regra é ativada ou desativada.

Atributos

<activationTime> não possui atributos.

Contido em

<activationTime> está contido no seguinte elemento:

- <activationInterval>

Contém

<activationTime> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 19. Elementos Contidos no Elemento <activationTime>

Elemento	Obrigatório ou opcional?
<start>	Opcional. 0 ou 1 ocorrência é permitida.
<stop>	Opcional. 0 ou 1 ocorrência é permitida.

Elemento after

O elemento <after> especifica a duração de tempo que a regra permanecerá ativa depois que se tornar ativa. Depois desse tempo, a regra deverá ser desativada.

Atributos

<after> possui os seguintes atributos:

Tabela 20. Atributos do Elemento <after>

Nome	Descrição	Tipo de dado	Obrigatório?
duration	Especifica o período da duração. O tipo de dado desse atributo depende do valor do atributo de unidade.	<ul style="list-style-type: none">• Se o valor do atributo de unidade for ISO-8601, o tipo de dado será xsd:duration.• Se o valor do atributo de unidade for milliseconds, o tipo de dado será xsd:positiveInteger.	Sim
unit	Especifica a unidade de tempo a ser utilizada. Os valores válidos para esse atributo são: <ul style="list-style-type: none">• ISO-8601• milliseconds	xsd:string	Sim

O uso do ISO 8601 padrão para duração de tempo

A codificação ISO-8601 como o valor para o atributo de unidade indica que o valor do atributo de duração é codificado de acordo com o padrão ISO 8601 para especificar uma duração de tempo como uma cadeia. A especificação de tipo de dado do esquema XML padrão utiliza ISO 8601 para fornecer um tipo de dado chamado duration. Esse tipo de dado é descrito em detalhes em <http://www.w3.org/TR/xmlschema-2/#duration>.

O formato do tipo de dado duration no esquema XML padrão é a seguinte cadeia:

PnYnMnDTnHnMnS

- *P* é o caractere que sempre começa a cadeia.
- *nY* representa o número de anos. Um ano é o mesmo que 365 dias. Portanto, a codificação 1Y é o mesmo que a codificação 365D.
- *nM* representa o número de meses. Um mês é o mesmo que 30 dias. Portanto, a codificação 1M é o mesmo que a codificação 30D.
- *nD* representa o número de dias.
- *T* é um separador que separa as unidades de dias (anos, meses e dias) das unidades de tempo (horas, minutos e segundos). As unidades de tempo sempre seguem *T*.
- *nH* representa o número de horas.
- *nM* representa o número de minutos.
- *nS* representa o número de segundos.

A seguir, exemplos do formato:

- P5DT12H é 5,5 dias.
- PT59M59S é 59 minutos e 59 segundos.
- P1M é 1 mês.

Contido em

<after> está contido no seguinte elemento:

- <stop>

Contém

<after> não contém elementos.

Elemento attributeAlias

O elemento <attributeAlias> fornece um nome de alias para associar atributos de eventos com o mesmo significado, mas com nomes diferentes em eventos diferentes. Por exemplo, três eventos diferentes podem utilizar estes três nomes diferentes para um atributo de evento que indique o nome do sistema que origina o evento: host, nome do host e origem. O elemento <attributeAlias> contém os elementos <eventAttribute> que descrevem os atributos de evento individuais que devem ser associados como um atributo de evento para a chave de agrupamento.

Detalhes

O elemento <attributeAlias> e seu atributo aliasName são válidos apenas no contexto de uma chave de agrupamento. Esse elemento e seu atributo não podem ser referenciados em nenhuma expressão, incluindo uma contida no elemento <computedValue>.

Atributos

<attributeAlias> possui o seguinte atributo:

Tabela 21. Atributos do Elemento <attributeAlias>

Nome	Descrição	Tipo de dado	Obrigatório?
aliasName	Define o nome dos atributos de eventos descritos nos elementos <eventAttribute> e que devem ser associados como um atributo de evento para a chave de agrupamento. Esse nome deve ser exclusivo na regra.	xsd:NMTOKEN	Sim

Contido em

<attributeAlias> está contido no seguinte elemento:

- <groupingKey>

Contém

<attributeAlias> contém o seguinte elemento:

Tabela 22. Elementos Contidos no Elemento <attributeAlias>

Elemento	Obrigatório ou opcional?
<eventAttribute>	2 ocorrências desse elemento são obrigatórias. Ocorrências adicionais são permitidas.

Elemento attributeName

O elemento <attributeName> contém o nome de um atributo de evento específico que faz parte da chave de agrupamento. Esse nome deve corresponder ao nome utilizado na chamada de método `getAttribute` da variável `act_event`.

Atributos

<attributeName> não possui atributos.

Contido em

<attributeName> está contido no seguinte elemento:

- <groupingKey>

Contém

<attributeName> não contém elementos.

Elemento booleanThreshold

O elemento <booleanThreshold> é válido apenas para a regra de limite. Ele contém uma expressão que é chamada conforme cada evento é recebido. A expressão calcula ou compara o valor do limite com base no evento atual e em quaisquer outros eventos que tenham sido aceitos pela regra. A expressão retorna um valor booleano de `true` ou `false` para indicar se o limite foi atendido.

Detalhes

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

<booleanThreshold> possui o seguinte atributo:

Tabela 23. Atributos do Elemento <booleanThreshold>

Nome	Descrição	Tipo de dado	Obrigatório?
expressionLanguage	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é java.	xsd:NMTOKEN	Sim

Contido em

<booleanThreshold> está contido no seguinte elemento:

- <thresholdRule>

Contém

<booleanThreshold> não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento collectionRule

O elemento <collectionRule> define uma regra de acordo com o padrão de coleta.

Atributos

<collectionRule> possui os seguintes atributos:

Tabela 24. Atributos do Elemento <collectionRule>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim

Tabela 24. Atributos do Elemento <collectionRule> (continuação)

Nome	Descrição	Tipo de dado	Obrigatório?
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não

Contido em

<collectionRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<collectionRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 25. Elementos Contidos no Elemento <collectionRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<eventSelector>	Opcional. 0 ou 1 ocorrência é permitida.
<groupingKey>	Opcional. 0 ou 1 ocorrência é permitida.
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onTimeWindowComplete>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Coleta” na página 11

A regra de coleta é definida pelo padrão de coleta. Ela coleta um grupo de eventos selecionados em um intervalo de tempo. É uma regra com preservação de estado.

Elemento comment

O elemento <comment> pode conter uma descrição da função e do objetivo de seu conjunto de regras, bloco de regra, regra ou variável contidos.

Atributos

<comment> não possui atributos.

Contido em

<comment> está contido nos seguintes elementos:

- <ruleSet>
- <ruleBlock>
- <collectionRule>
- <computationRule>
- <duplicateRule>
- <filterRule>
- <sequenceRule>
- <thresholdRule>
- <timerRule>
- <variable>

Contém

<comment> não contém elementos.

Elemento computationRule

O elemento <computationRule> define uma regra de acordo com o padrão de computação.

Atributos

<computationRule> possui os seguintes atributos:

Tabela 26. Atributos do Elemento <computationRule>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não

Contido em

<computationRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<computationRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 27. Elementos Contidos no Elemento <computationRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<eventSelector>	Opcional. 0 ou 1 ocorrência é permitida.
<groupingKey>	Opcional. 0 ou 1 ocorrência é permitida.
<computeFunction>	Obrigatório. Apenas 1 ocorrência é permitida.
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onTimeWindowComplete>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Computação” na página 12

Uma regra de computação é definida pelo padrão de computação. Ela aplica um cálculo (por meio de uma expressão) nos eventos coletados conforme cada evento é recebido em um intervalo de tempo. É uma regra com preservação de estado.

Elemento computedThreshold

O elemento <computedThreshold> é válido apenas para a regra de limite. Ele contém uma expressão que é chamada conforme cada evento é recebido e que calcula o valor do limite com base no evento atual e em quaisquer outros eventos que tenham correspondido aos critérios de seleção de evento da regra. A expressão retorna o valor do limite calculado para ser armazenado em uma variável definida para a regra. Em seguida, a regra utiliza o valor do limite calculado para comparar com o valor do limite definido.

Detalhes

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

<computedThreshold> possui os seguintes atributos:

Tabela 28. Atributos do Elemento <computedThreshold>

Nome	Descrição	Tipo de dado	Obrigatório?
expressionLanguage	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é java.	xsd:NMTOKEN	Sim

Tabela 28. Atributos do Elemento <computedThreshold> (continuação)

Nome	Descrição	Tipo de dado	Obrigatório?
limite	Define o valor do limite que deve ser atendido. Esse valor de limite definido deve ser uma representação em cadeia de um valor numérico que pode ser convertido em um tipo de dado que é válido para a variável da regra.	xsd:string	Sim
assignTo	Identifica o nome da variável que contém o valor do limite calculado que é retornado dessa expressão. Essa variável já deverá estar definida para a regra (no conjunto de regras, bloco de regra ou nível de regra) utilizando o elemento <variable>. Ela deve ser definida com um dos seguintes tipos de dados numéricos: <ul style="list-style-type: none"> • java.lang.Double • java.lang.Float • java.lang.Integer • java.lang.Long • java.lang.String Se a variável for definida no nível do conjunto de regras ou do bloco de regra, ela não será reinicializada depois que o padrão de regra for correspondido.	xsd:NMTOKEN	Sim
thresholdComparison	Define o operador para comparar o valor do limite calculado com o valor do limite definido. Os valores válidos para esse operador são: <ul style="list-style-type: none"> • lessThan • lessThanOrEqualTo • greaterThan • greaterThanOrEqualTo 	xsd:string	Sim

Contido em

<computedThreshold> está contido no seguinte elemento:

- <thresholdRule>

Contém

<computedThreshold> não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o

mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento `computedValue`

O elemento `<computedValue>` contém uma expressão executada quando a regra recebe um evento para criar um valor de cadeia que se baseia no valor de um ou de mais atributos do evento. Esse valor de cadeia pode então ser utilizado na chave de agrupamento.

Detalhes

Às vezes, um autor de regras pode querer utilizar itens como estes na chave de agrupamento:

- Uma subcadeia de um valor de atributo do evento. Por exemplo, se um valor de atributo do evento contiver um endereço IP incorporado, a expressão no elemento `<computedValue>` poderia extrair esse endereço IP como valor exclusivo para utilizar na chave de agrupamento.
- Subcadeias dos valores de vários atributos de evento diferentes. Por exemplo, a expressão no elemento `<computedValue>` poderia extrair as subcadeias e combiná-las para criar um valor exclusivo para utilizar na chave de agrupamento.

Se a expressão no elemento `<computedValue>` retornar um valor nulo, a regra tratará este como um valor de atributo ausente.

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

`<computedValue>` possui o seguinte atributo:

Tabela 29. Atributos do Elemento `<computedValue>`

Nome	Descrição	Tipo de dado	Obrigatório?
<code>expressionLanguage</code>	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é <code>java</code> .	<code>xsd:NMTOKEN</code>	Sim

Contido em

`<computedValue>` está contido no seguinte elemento:

- `<groupingKey>`

Contém

`<computedValue>` não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento `computeFunction`

O elemento `<computeFunction>` é válido apenas para a regra de computação. Ele contém uma expressão que é chamada conforme cada evento é recebido e que retorna um valor a ser armazenado em uma variável que é definida para a regra. O valor a ser retornado dessa expressão deve corresponder ao tipo de dado da variável que é nomeada no atributo `assignTo` do elemento `<computeFunction>`.

Detalhes

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

`<computeFunction>` possui os seguintes atributos:

Tabela 30. Atributos do Elemento `<computeFunction>`

Nome	Descrição	Tipo de dado	Obrigatório?
<code>expressionLanguage</code>	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é <code>java</code> .	<code>xsd:NMTOKEN</code>	Sim
<code>assignTo</code>	Identifica o nome da variável que contém o valor que é retornado dessa expressão. Essa variável já deverá estar definida para a regra (no conjunto de regras, bloco de regra ou nível de regra) utilizando o elemento <code><variable></code> . Se a variável for definida no nível do conjunto de regras ou do bloco de regra, ela não será reinicializada depois que o padrão de regra for correspondido.	<code>xsd:NMTOKEN</code>	Sim

Contido em

`<computeFunction>` está contido no seguinte elemento:

- `<computationRule>`

Contém

`<computeFunction>` não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento `dateTime`

O elemento `<dateTime>` especifica a data e a hora em que uma regra é ativada ou desativada. No entanto, a regra será ativada ou desativada somente se tiver sido carregada em um mecanismo em execução da Tecnologia ACT antes dessa hora especificada.

Detalhes

Se a regra não tiver sido carregada em um mecanismo em execução da Tecnologia ACT antes da hora especificada para ativação, a regra nunca será ativada. Se a regra não tiver sido carregada em um mecanismo em execução da Tecnologia ACT antes da hora especificada para desativação, ela será configurada com o estado definido pelo elemento `<start>` e nunca será desativada pelo elemento `<stop>`.

O conteúdo do elemento `<dateTime>` deve ser uma cadeia que siga o formato do tipo de dado `dateTime` no esquema XML padrão. Por exemplo, `dateTime` consiste em seqüências de caracteres de comprimento finito na seguinte forma:

`yyyy '-' mm '-' dd 'T' hh ':' mm ':' ss ('.' s+)? (zzzzzz)?`

- `yyyy` é um numeral com quatro ou mais dígitos que representa o ano. Se tiver mais de quatro dígitos, os zeros à esquerda serão proibidos e `0000` será proibido.
- Os `'-'`s restantes são separadores entre as partes da data.
- O primeiro `mm` é um numeral de dois dígitos que representa o mês, iniciando com `01`.
- `dd` é um numeral de dois dígitos que representa o dia do mês, iniciando com `01`.
- `T` é um separador para indicar que a hora do dia vem em seguida.
- `hh` é um numeral de dois dígitos que representa a hora do dia no sistema de 24 horas, iniciando com `00` e terminando com `23`.
- `:` é um separador entre as partes da hora do dia.
- O segundo `mm` é um numeral de dois dígitos que representa o minuto, iniciando com `00` e terminando com `59`.
- `ss` é um numeral de dois dígitos que representa o total de segundos, iniciando com `00` e terminando com `59`.
- `'.' s+`, se presente, representa os segundos fracionários.
- `zzzzzz`, se presente, representa o fuso horário. O fuso horário consiste em seqüências de caracteres de comprimento finito na forma `(('+' | '-') hh ':' mm) | 'Z'`, em que:
 - `'+'`, se presente, representa uma duração não negativa e `'-'` não deve estar presente.
 - `'-'`, se presente, representa uma duração não positiva e `'+'` não deve estar presente.
 - `hh` é um numeral de dois dígitos que representa as horas, iniciando com `00` e terminando com `14`.

- *mm* é um numeral de dois dígitos que representa os minutos, iniciando com 00 e terminando com 59. No entanto, se o valor de horas for 14, o valor de minutos deverá ser 00
- Z é a estenografia de UTC (+00:00 ou -00:00) e assim, nenhum outro elemento de fuso horário deverá estar presente.

A seguir, dois exemplos do conteúdo do elemento <dateTime>:

- 2005-06-01T13:05:06.07 é 1 de junho de 2005, às 13h05, 6 segundos e 7 centésimos na hora local.
- 2005-06-01T13:05:06.07Z é 1 de junho de 2005, às 13h05, 6 segundos e 7 centésimos na hora UTC, que seria 1 de junho de 2005 às 9h05, 6 segundos e 7 centésimos EDT (ou 2005-06-01T09:05:06.07-04:00)

Atributos

<dateTime> não possui atributos.

Contido em

<dateTime> está contido nos seguintes elementos:

- <start>
- <stop>

Contém

<dateTime> não contém elementos.

Elemento deactivateOnEvent

O elemento <deactivateOnEvent> define os eventos que podem deativar a regra ou, para regras definidas com um elemento <groupingKey>, uma instância de regra.

A seguir, as três maneiras possíveis de seleção de eventos:

- A utilização de um ou mais elementos <eventType> com um elemento <filteringPredicate>
- A utilização de um ou mais elementos <eventType> sem um elemento <filteringPredicate>
- A utilização de um elemento <filteringPredicate> sem nenhum elemento <eventType>

Se a regra estiver ativa e nenhum elemento <eventType> ou <filteringPredicate> estiver codificado, qualquer evento que ocorrer será selecionado.

A não codificação de nenhum elemento <eventType> pode afetar negativamente o desempenho do sistema.

Suponha que você queira selecionar todos os eventos do tipo Falha na Auditoria. É possível utilizar um predicado de filtragem para aperfeiçoar ainda mais os critérios de seleção para incluir apenas os eventos que possuam um atributo de evento com um determinado valor. Por exemplo, você codificaria um elemento <eventType> para selecionar todos os eventos do tipo Falha na Auditoria e codificaria um elemento <filteringPredicate> para selecionar apenas os eventos que possuem um atributo de nome do host com o valor MyCriticalSystem.

Atributos

<deactivateOnEvent> não possui atributos.

Contido em

<deactivateOnEvent> está contido nos seguintes elementos:

- <activationInterval>
- <activationByGroupingKey>

Contém

<deactivateOnEvent> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 31. Elementos Contidos no Elemento <deactivateOnEvent>

Elemento	Obrigatório ou opcional?
<eventType>	Opcional. 0 ou mais ocorrências são permitidas.
<filteringPredicate>	Opcional. 0 ou 1 ocorrência é permitida.

Elemento duplicateRule

O elemento <duplicateRule> define uma regra de acordo com o padrão de duplicata.

Atributos

<duplicateRule> possui os seguintes atributos:

Tabela 32. Atributos do Elemento <duplicateRule>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não

Contido em

<duplicateRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<duplicateRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 33. Elementos Contidos no Elemento <duplicateRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<eventSelector>	Opcional. 0 ou 1 ocorrência é permitida.
<groupingKey>	Opcional. 0 ou 1 ocorrência é permitida.
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onDetection>	Opcional. 0 ou 1 ocorrência é permitida.
<onNextEvent>	Opcional. 0 ou 1 ocorrência é permitida.
<onTimeWindowComplete>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Duplicata” na página 12

Uma regra de duplicata é definida pelo padrão de duplicata. Ela conta o segundo e os subseqüentes eventos que são aceitos no intervalo de tempo especificado, mas ignora o processamento do conjunto de regras deles. É uma regra com preservação de estado.

Elemento eventAttribute

O elemento <eventAttribute> fornece uma maneira de associar um tipo de evento e um atributo de evento como parte do nome do alias do atributo definido pelo elemento <attributeAlias>.

Atributos

<eventAttribute> possui os seguintes atributos:

Tabela 34. Atributos do Elemento <eventAttribute>

Nome	Descrição	Tipo de dado	Obrigatório?
type	Define o nome do tipo de evento. É o mesmo nome utilizado para o atributo type no elemento <eventType>.	xsd:NMTOKEN	Sim
attributeName	Especifica o nome completo do atributo de evento que está sendo associado a outros atributos de evento por meio do nome do alias do atributo. Esse nome deve corresponder ao nome utilizado na variável act_event para chamar o método getAttribute.	xsd:string	Sim

Contido em

<eventAttribute> está contido no seguinte elemento:

- <attributeAlias>

Contém

<eventAttribute> não contém elementos.

Elemento eventCountThreshold

O elemento <eventCountThreshold> é válido apenas para a regra de limite. Ele define o número de eventos que devem atender aos critérios de seleção de eventos em um determinado período de tempo. O elemento <eventCountThreshold> também especifica um de dois possíveis modos de intervalo de tempo, fixo ou móvel, para o espaço de tempo.

Detalhes

intervalo fixo

Esse intervalo começa quando o primeiro evento que atende aos critérios de seleção de eventos é recebido e termina quando ocorre uma das seguintes situações:

- A regra atende a seu limite na duração de tempo especificada.
- A duração de tempo especificada passou.

intervalo móvel

Esse intervalo começa quando o primeiro evento que atende aos critérios de seleção de eventos é recebido. No entanto, quando a regra não atende a seu limite e a duração de tempo especificada passa, o espaço de tempo ajusta (move) a hora de início para a hora de recepção do evento de um novo “primeiro” evento, que normalmente é o próximo evento que for aceito. O intervalo móvel continua a ser ajustado assim até que ocorra uma das seguintes situações:

- A regra atende a seu limite na duração de tempo especificada.
- Após a recepção do evento que começa o espaço de tempo, nenhum evento subsequente é recebido durante o tempo especificado.

O evento que começa o espaço de tempo (torna-se o novo “primeiro” evento) é aquele com uma hora de recepção que atende a estes critérios: a hora da recepção, somada à duração do intervalo de tempo da regra, é maior que a hora atual. A seguir, os critérios na forma de uma equação:

hora de recepção do evento + duração do intervalo de tempo da regra > hora atual

Quando esse evento não existir, o intervalo móvel não poderá mais ajustar a hora e o intervalo será encerrado.

A regra de limite conta cada evento aceito até que o limite seja atendido ou o período de tempo seja encerrado. Em seguida, ele executa as ações definidas no elemento <onDetection> ou no elemento <onTimeOut>, conforme apropriado.

ações de <onDetection>

Essas ações são executadas quando a contagem de eventos é igual ao valor definido pelo atributo de limite do elemento <eventCountThreshold>, indicando que o limite foi atendido.

ações de <onTimeOut>

A execução dessas ações depende de o modo de intervalo de tempo ser fixo ou móvel.

modo fixo

Nesse modo, essas ações são executadas quando o espaço de tempo expira.

modo móvel

Nesse modo, essas ações serão executadas se, após a recepção do evento que começa o espaço de tempo, nenhum evento subsequente for recebido durante o tempo especificado. Ou seja, nenhum evento será recebido com uma hora de recepção que, quando somada à duração do intervalo de tempo da regra, for maior que a hora atual.

O modo de intervalo de tempo do espaço de tempo é definido pelo atributo `timeIntervalMode` do elemento `<eventCountThreshold>`. O cenário a seguir ilustra o comportamento e as diferenças entre os dois possíveis modos de intervalo de tempo.

Cenário que ilustra os Modos Fixo e Móvel

Suponha que a regra receba quatro eventos que atendam aos critérios de seleção de eventos, um evento em cada um destes horários: 8h, 8h04, 8h06 e 8h07. O limite de contagem do evento é 3 e a duração do espaço de tempo é de 5 minutos.

Comportamento da regra com o modo fixo

Nesse modo de intervalo de tempo, a regra de limite começa a ser processada às 8h e executa as ações de `<onTimeOut>` às 8h05 porque recebe apenas 2 eventos em 5 minutos. Portanto, ela não atende a seu limite no espaço de tempo. Quando o terceiro evento é recebido às 8h06, a regra de limite começa a ser processada novamente e executa as ações de `<onTimeOut>` às 8h11 porque recebe apenas 2 eventos em 5 minutos.

O modo fixo é estático.

Comportamento da regra com o modo móvel

Nesse modo de intervalo de tempo, a regra de limite começa a ser processada às 8h. Às 8h05, quando está planejada a conclusão do espaço de tempo, a regra determina que recebeu apenas 2 eventos. A regra descarta então o evento recebido às 8h e recalcula para que a duração seja encerrada às 8h09 (desde que o primeiro evento seja agora aquele recebido às 8h04). Quando a regra recebe o evento às 8h07, ela executa as ações de `<onDetection>` porque agora atendeu a seu limite (3 eventos, às 8h04, 8h06 e 8h07) dentro do último espaço de tempo (8h04 – 8h09).

O modo móvel é dinâmico porque continua a ajustar (mover) a hora inicial na tentativa de atender a seu limite dentro do espaço de tempo.

Suponha agora que a regra receba quatro eventos que atendam aos critérios de seleção de eventos, um evento em cada um destes horários: 8h, 8h04, 8h06 e 8h10. O limite de contagem do evento é 3 e a duração do espaço de tempo é de 5 minutos.

Comportamento da regra com o modo móvel

Nesse caso, a regra de limite começa a ser processada às 8h. Às 8h05, quando está planejada a conclusão do espaço de tempo, a regra determina que recebeu apenas 2 eventos. A regra descarta então o evento recebido às

8h e recalcula para que a duração seja encerrada às 8h09 (desde que o primeiro evento seja agora aquele recebido às 8h04).

Às 8h09, quando está planejada a conclusão do espaço de tempo, a regra determina que recebeu apenas 2 eventos. A regra descarta então o evento recebido às 8h04 e recalcula para que a duração seja encerrada às 8h11 (desde que o primeiro evento seja agora aquele recebido às 8h06).

Às 8h11, quando está planejada a conclusão do espaço de tempo, a regra determina que recebeu apenas 2 eventos. A regra descarta então o evento recebido às 8h06 e recalcula para que a duração seja encerrada às 8h15 (desde que o primeiro evento seja agora aquele recebido às 8h10).

Às 8h15, quando está planejada a conclusão do espaço de tempo, a regra determina que não recebeu eventos desde o evento às 8h10 que começou o espaço de tempo. A regra executa então as ações de <onTimeout>.

Atributos

<eventCountThreshold> possui os seguintes atributos:

Tabela 35. Atributos do Elemento <eventCountThreshold>

Nome	Descrição	Tipo de dado	Obrigatório?
limite	Define o número de eventos que devem atender aos critérios de seleção de eventos em um determinado período de tempo. Esse é o limite de contagem de eventos que deve ser atendido. O valor tem que ser um número inteiro positivo.	xsd:positiveInteger	Sim
timeIntervalMode	Define se o intervalo de tempo para o espaço de tempo é fixo ou móvel. Os valores válidos para esse atributo são: <ul style="list-style-type: none">• fixed (o valor padrão)• sliding	xsd:string	Não

Contido em

<eventCountThreshold> está contido no seguinte elemento:

- <thresholdRule>

Contém

<eventCountThreshold> não contém elementos.

Elemento eventSelector

O elemento <eventSelector> define os eventos selecionados para processamento por uma regra.

Detalhes

A seguir, as três maneiras possíveis de seleção de eventos:

- A utilização de um ou mais elementos <eventType> com um elemento <filteringPredicate>
- A utilização de um ou mais elementos <eventType> sem um elemento <filteringPredicate>
- A utilização de um elemento <filteringPredicate> sem nenhum elemento <eventType>

Em casos especiais em que você deseja que uma regra processe todos os eventos, estão disponíveis as seguintes opções:

- Não codificar um elemento <eventSelector>.
- Codificar um elemento <eventSelector> que não contenha elementos.

A não codificação de nenhum elemento <eventType> pode afetar negativamente o desempenho do sistema.

Suponha que você queira selecionar todos os eventos do tipo Falha na Auditoria. É possível utilizar um predicado de filtragem para aperfeiçoar ainda mais os critérios de seleção para incluir apenas os eventos que possuam um atributo de evento com um determinado valor. Por exemplo, você codificaria um elemento <eventType> para selecionar todos os eventos do tipo Falha na Auditoria e codificaria um elemento <filteringPredicate> para selecionar apenas os eventos que possuem um atributo de nome do host com o valor MyCriticalSystem.

Atributos

<eventSelector> possui o seguinte atributo:

Tabela 36. Atributos do Elemento <eventSelector>

Nome	Descrição	Tipo de dado	Obrigatório?
alias	Esse atributo é válido apenas em uma regra de sequência, que é a única regra que possui vários elementos <eventSelector>. Ela nomeia exclusivamente um evento que é selecionado por um determinado seletor de eventos na regra de sequência. Os predicados e as ações de filtragem podem então utilizar esse nome de alias para acessar esse evento.	xsd:NMTOKEN	Não

Contido em

<eventSelector> está contido nos seguintes elementos:

- <collectionRule>
- <computationRule>
- <duplicateRule>
- <filterRule>
- <sequenceRule>
- <thresholdRule>

Contém

<eventSelector> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 37. Elementos Contidos no Elemento <eventSelector>

Elemento	Obrigatório ou opcional?
<eventType>	Opcional. 0 ou mais ocorrências são permitidas.
<filteringPredicate>	Opcional. 0 ou 1 ocorrência é permitida.

Elemento eventType

O elemento <eventType> define o tipo de evento selecionado para processamento por uma regra ou que ativa ou desativa a regra.

Atributos

<eventType> possui o seguinte atributo:

Tabela 38. Atributos do Elemento <eventType>

Nome	Descrição	Tipo de dado	Obrigatório?
type	Define o tipo de evento. Para eventos que estão em conformidade com a especificação Common Base Event, esse nome é o valor do atributo extensionName. No caso de eventos do IBM Tivoli Enterprise Console, esse é o nome da classe de evento definido no arquivo BAROC. Os eventos baseados em outros formatos podem utilizar um atributo diferente para especificar o tipo de evento.	xsd:NMTOKEN	Sim

Contido em

<eventType> está contido nos seguintes elementos:

- <activateOnEvent>
- <deactivateOnEvent>
- <eventSelector>

Contém

<eventType> não contém elementos.

Elemento `filteringPredicate`

O elemento `<filteringPredicate>` contém uma expressão que restringe ainda mais os eventos selecionados para processamento pela regra ou selecionados para ativar ou desativar a regra. Portanto, os eventos podem ser filtrados de forma ainda mais abrangente do que filtrar apenas pelo tipo de evento por meio do elemento `<eventType>`.

Detalhes

A expressão define uma condição e retorna um valor booleano, `true` (a condição é atendida) ou `false` (a condição não é atendida).

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

`<filteringPredicate>` possui o seguinte atributo:

Tabela 39. Atributos do Elemento `<filteringPredicate>`

Nome	Descrição	Tipo de dado	Obrigatório?
<code>expressionLanguage</code>	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é <code>java</code> .	<code>xsd:NMTOKEN</code>	Sim

Contido em

`<filteringPredicate>` está contido nos seguintes elementos:

- `<activateOnEvent>`
- `<deactivateOnEvent>`
- `<eventSelector>`

Contém

`<filteringPredicate>` não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento `filterRule`

O elemento `<filterRule>` define uma regra de acordo com o padrão de filtragem.

Atributos

<filterRule> possui os seguintes atributos:

Tabela 40. Atributos do Elemento <filterRule>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não

Contido em

<filterRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<filterRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 41. Elementos Contidos no Elemento <filterRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<eventSelector>	Opcional. 0 ou 1 ocorrência é permitida.
<onDetection>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Filtragem” na página 13

Uma regra de filtragem é definida pelo padrão de filtragem. Ela executa uma determinada ação ao aceitar um evento. Ela age apenas em um único evento e é, portanto, uma regra sem preservação de estado.

Elemento groupingKey

Normalmente, cada regra ativa possui uma instância de regra, ou cópia, em execução no mecanismo da Tecnologia ACT. No entanto, às vezes, a mesma regra é necessária para grupos diferentes de eventos, relacionados, com frequência, a diferentes grupos de recursos. A chave de agrupamento é um ou mais atributos de eventos, ou partes de atributos de eventos, que podem ser utilizados para separar os eventos selecionados em diferentes grupos para processamento exclusivo como um grupo. O elemento <groupingKey> define a chave de agrupamento de uma regra. A finalidade do elemento <groupingKey> é direcionar a regra para criar uma instância de regra separada (ou uma cópia de si mesma) para cada grupo de eventos que compartilham características comuns (conforme definido pelos valores de seus atributos que compõem a chave de agrupamento).

Detalhes

Os dois cenários a seguir ilustram a importância da chave de agrupamento.

Cenário 1:

Ocorrem dois eventos, um evento DB2down e um evento DB2up. O programa DB2 é executado em três computadores denominados A, B e C. Uma regra de sequência é definida para correlacionar um evento DB2down dentro de um evento DB2up e para alertar o operador quando o programa DB2 pára e não reinicia.

Se a regra de sequência for definida sem a chave de agrupamento e um evento DB2down for recebido do computador A, um evento DB2up de qualquer um dos computadores completará a sequência, mas não realizará o que foi planejado. No entanto, se a chave de agrupamento tiver sido definida como o atributo de nome do host, uma cópia ou instância exclusiva da regra seria criada para cada valor exclusivo do atributo de nome do host nos eventos selecionados. O mecanismo da Tecnologia ACT enviaria cada evento para a instância de regra correta (a instância de regra do valor de nome do host desse evento). Portanto, se um evento DB2down for recebido do computador A, o mecanismo da Tecnologia ACT criará uma instância de regra para o computador A. Se um evento DB2down for recebido do computador B, o mecanismo da Tecnologia ACT criará uma segunda instância de regra para o computador B. Quando um evento DB2up for recebido do computador B, o mecanismo da Tecnologia ACT processará esse evento na segunda instância da regra. A sequência estará completa e o operador será alertado porque os eventos DB2down e DB2up do computador B estão correlacionados corretamente.

Cenário 2:

Ocorre um evento para uma mensagem Tentativa incorreta de login em todos os computadores de um determinado ambiente. O evento contém um ID de usuário. É definida uma regra de limite para emitir um aviso ao operador se esse evento ocorrer mais de 10 vezes em 5 minutos.

Uma chave de agrupamento poderia ser definida como o ID de usuário. Em seguida, uma nova instância de regra seria criada para cada ID de usuário exclusivo. As tentativas de login de cada usuário seriam acompanhadas em uma instância de regra de limite exclusiva, com cada instância tendo uma conta separada do número de tentativas de login desse usuário. O operador receberia um aviso se algum ID de usuário excedesse 10 logins incorretos em 5 minutos.

Outras variações dessa idéia incluem:

- A chave de agrupamento poderia ser definida como o nome do host, em vez de o ID de usuário. Essa opção poderia detectar um grande número de tentativas incorretas de login em um único computador.
- A chave de agrupamento poderia ser definida como uma combinação do nome do host e do ID de usuário. Essa opção poderia detectar tentativas potenciais de hack por um ID de usuário específico em um computador específico.

Se o mesmo atributo estiver em todos os tipos de eventos especificados para uma regra, o uso do elemento <attributeName> será a maneira mais simples e mais comum de definir uma chave de agrupamento.

Atributos

<groupingKey> possui o seguinte atributo:

Tabela 42. Atributos do Elemento <groupingKey>

Nome	Descrição	Tipo de dado	Obrigatório?
missingAttributeHandling	<p>Define a ação que a regra deve executar sob uma destas condições:</p> <ul style="list-style-type: none"> • Quando um evento selecionado possuir um atributo que participa da chave de agrupamento, mas o valor desse atributo está ausente • Quando a expressão no elemento <computedValue> retornar um valor nulo. A regra tratará este valor nulo como um valor de atributo ausente. <p>Os valores válidos para o atributo missingAttributeHandling são:</p> <ul style="list-style-type: none"> • ignoreEvent (o valor padrão), significando que a regra ignorará o evento e não executará ação nele. • ignoreAttribute, significando que a regra aceitará o evento, mas ignorará o atributo com o valor ausente. O mecanismo da Tecnologia ACT incluirá, então, um valor substituto para o atributo. 	xsd:string	Não

Contido em

<groupingKey> está contido nos seguintes elementos:

- <collectionRule>
- <computationRule>
- <duplicateRule>
- <sequenceRule>

- <thresholdRule>

Contém

<groupingKey> contém os elementos a seguir.

Tabela 43. Elementos Contidos no Elemento <groupingKey>

Elemento	Obrigatório ou opcional?
<attributeAlias>	1 desses elementos é obrigatório. Codificar mais de 1 desses elementos é opcional. São permitidas várias ocorrências dos três elementos. Esses elementos podem ser codificados em qualquer ordem.
<attributeName>	
<computedValue>	

Elemento import

O elemento <import> contém uma expressão que especifica os módulos externos (por exemplo, classes Java) que serão importados para serem utilizados em outras expressões nas regras.

Detalhes

O código de expressão é uma cadeia no elemento <import>. O compilador da Tecnologia ACT utiliza as instruções de importação fornecidas pelos elementos <import> para compilar código de expressão nas regras que chamam métodos externos.

Atributos

<import> possui o seguinte atributo:

Tabela 44. Atributos do Elemento <import>

Nome	Descrição	Tipo de dado	Obrigatório?
expressionLanguage	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é java.	xsd:NMTOKEN	Sim

Contido em

<import> está contido nos seguintes elementos:

- <ruleSet>
- <ruleBlock>

Contém

<import> não contém elementos.

Conceitos relacionados

“Importando e Acessando Módulos e Objetos Externos” na página 21

Este exemplo indica como tornar código externo (por exemplo, classes Java) e objetos externos acessíveis às expressões. Objeto externo é aquele que o aplicativo cria para se comunicar com as expressões.

Elemento `inactiveWhenLoaded`

O elemento `<inactiveWhenLoaded>` especifica que uma regra ficará inativa quando for carregada pelo mecanismo da Tecnologia ACT. A regra permanecerá inativa até que seja ativada por outro meio.

Atributos

`<inactiveWhenLoaded>` não possui atributos.

Contido em

`<inactiveWhenLoaded>` está contido no seguinte elemento:

- `<start>`

Contém

`<inactiveWhenLoaded>` não contém elementos.

Elemento `lifeCycleActions`

O elemento `<lifeCycleActions>` contém elementos que definem as ações a serem executadas nos quatro estágios primários no ciclo de vida de uma regra.

Detalhes

As ações definidas para os estágios de carregamento e ativação são chamadas depois que a regra é realmente carregada ou ativada, mas antes de começar qualquer processamento. As ações definidas para os estágios de desativação e descarregamento são chamadas imediatamente antes que a regra seja realmente desativada ou descarregada.

Atributos

`<lifeCycleActions>` não possui atributos.

Contido em

`<lifeCycleActions>` está contido nos seguintes elementos:

- `<collectionRule>`
- `<computationRule>`
- `<duplicateRule>`
- `<filterRule>`
- `<sequenceRule>`
- `<thresholdRule>`
- `<timerRule>`

Contém

`<lifeCycleActions>` contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 45. Elementos Contidos no Elemento <lifeCycleActions>

Elemento	Obrigatório ou opcional?
<onLoad>	Opcional. 0 ou 1 ocorrência é permitida.
<onActivation>	Opcional. 0 ou 1 ocorrência é permitida.
<onDeactivation>	Opcional. 0 ou 1 ocorrência é permitida.
<onUnload>	Opcional. 0 ou 1 ocorrência é permitida.

Elemento never

O elemento <never> especifica que uma regra nunca será desativada em um determinado tempo. Uma regra ainda poderá ser desativada por um evento ou por outro meio.

Atributos

<never> não possui atributos.

Contido em

<never> está contido no seguinte elemento:

- <stop>

Contém

<never> não contém elementos.

Elemento onActivation

O elemento <onActivation> especifica a ação ou o conjunto de ações a serem executadas quando a regra for ativada. A ação <onActivation> será chamada depois que a regra for ativada, mas antes de começar qualquer processamento.

Detalhes

Se o conjunto de regras contiver várias regras que forem ativadas na mesma data e hora ou pelo mesmo evento e que possuírem o mesmo espaço de tempo, as seguintes ações para essas regras não serão executadas exatamente ao mesmo tempo:

- Ações de resposta da regra nos elementos <onTimeout> e <onTimeWindowComplete>
- Ações de ciclo de vida nos elementos <onActivation> e <onDeactivation>

Essas ações são executadas em sequência em qualquer ordem. Elas não são executadas necessariamente na ordem em que são codificadas no conjunto de regras. Como cada ação deve ser concluída antes do início da próxima ação na sequência, as ações não são executadas ao mesmo tempo.

Atributos

<onActivation> não possui atributos.

Contido em

<onActivation> está contido no seguinte elemento:

- <lifeCycleActions>

Contém

<onActivation> contém o seguinte elemento:

Tabela 46. Elementos Contidos no Elemento <onActivation>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onDeactivation

O elemento <onDeactivation> especifica a ação ou o conjunto de ações a serem executadas quando a regra for desativada. A ação <onDeactivation> é chamada imediatamente antes de a regra ser desativada.

Detalhes

Se o conjunto de regras contiver várias regras que forem ativadas na mesma data e hora ou pelo mesmo evento e que possuírem o mesmo espaço de tempo, as seguintes ações para essas regras não serão executadas exatamente ao mesmo tempo:

- Ações de resposta da regra nos elementos <onTimeOut> e <onTimeWindowComplete>
- Ações de ciclo de vida nos elementos <onActivation> e <onDeactivation>

Essas ações são executadas em sequência em qualquer ordem. Elas não são executadas necessariamente na ordem em que são codificadas no conjunto de regras. Como cada ação deve ser concluída antes do início da próxima ação na sequência, as ações não são executadas ao mesmo tempo.

Atributos

<onDeactivation> não possui atributos.

Contido em

<onDeactivation> está contido no seguinte elemento:

- <lifeCycleActions>

Contém

<onDeactivation> contém o seguinte elemento:

Tabela 47. Elementos Contidos no Elemento <onDeactivation>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onDetection

O elemento <onDetection> é válido apenas para as regras de duplicata, de filtragem, de sequência e de limite. Ele especifica a ação ou o conjunto de ações a serem executadas quando o padrão de regra for detectado.

Detalhes

Tabela 48 descreve como o padrão de regra é detectado para cada tipo de regra em que uma ação <onDetection> é válida.

Tabela 48. Como um Padrão de Regra é Detectado com Base no Tipo de Regra

Tipo de regra	Como o padrão de regra é detectado
duplicata	Esse padrão de regra é detectado quando o primeiro evento que atende aos critérios de seleção de eventos é recebido.
filtragem	Esse padrão de regra é detectado quando qualquer evento que atende aos critérios de seleção de eventos é recebido.
sequência	Esse padrão de regra é detectado quando uma sequência de eventos que atendem aos critérios de seleção de eventos é recebida na ordem apropriada e dentro do espaço de tempo.
limite	Esse padrão de regra é detectado quando eventos que atendem aos critérios de seleção de eventos são recebidos dentro do espaço de tempo e o limite é atendido.

Atributos

<onDetection> não possui atributos.

Contido em

<onDetection> está contido nos seguintes elementos:

- <duplicateRule>
- <filterRule>
- <sequenceRule>
- <thresholdRule>

Contém

<onDetection> contém o seguinte elemento:

Tabela 49. Elementos Contidos no Elemento <onDetection>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onLoad

O elemento <onLoad> especifica a ação ou o conjunto de ações a serem executadas quando a regra for carregada (ou implementada) no mecanismo da Tecnologia ACT em execução. A ação <onLoad> será chamada depois que a regra for carregada, mas antes de começar qualquer processamento.

Atributos

<onLoad> não possui atributos.

Contido em

<onLoad> está contido no seguinte elemento:

- <lifeCycleActions>

Contém

<onLoad> contém este elemento:

Tabela 50. Elementos Contidos no Elemento <onLoad>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onNextEvent

O elemento <onNextEvent> é válido apenas para a regra de duplicata. Ele especifica a ação ou o conjunto de ações a serem executadas quando a regra de duplicata recebe o segundo e cada evento subsequente que atender aos critérios de seleção de eventos no espaço de tempo especificado.

Detalhes

Para regras de duplicata, o mecanismo da Tecnologia ACT ignora o processamento do conjunto de regras para o segundo e cada evento subsequente que corresponder aos critérios de seleção de eventos dentro do espaço de tempo especificado. Portanto, a única razão para codificar uma ação <onNextEvent> é especificar processamento alternativo para o segundo e cada evento subsequente.

Atributos

<onNextEvent> não possui atributos.

Contido em

<onNextEvent> está contido no seguinte elemento:

- <duplicateRule>

Contém

<onNextEvent> contém o seguinte elemento:

Tabela 51. Elementos Contidos no Elemento <onNextEvent>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onTimeOut

O elemento <onTimeOut> é válido apenas para as regras de seqüência e de limite. Ele especifica a ação ou o conjunto de ações a serem executadas se o espaço de tempo da regra expirar.

Detalhes

Tabela 52 descreve como o espaço de tempo expira para cada tipo de regra em que uma ação <onTimeOut> é válida.

Tabela 52. Como o Espaço de Tempo Expira com Base no Tipo de Regra

Tipo de regra	Como o espaço de tempo expira
seqüência	O espaço de tempo expirará se um ou mais eventos forem aceitos, mas a seqüência completa de eventos não for recebida dentro do espaço de tempo.
limite	O espaço de tempo expirará se um ou mais eventos forem aceitos, mas o limite não for atendido dentro do espaço de tempo.

Se o conjunto de regras contiver várias regras que forem ativadas na mesma data e hora ou pelo mesmo evento e que possuírem o mesmo espaço de tempo, as seguintes ações para essas regras não serão executadas exatamente ao mesmo tempo:

- Ações de resposta da regra nos elementos <onTimeOut> e <onTimeWindowComplete>
- Ações de ciclo de vida nos elementos <onActivation> e <onDeactivation>

Essas ações são executadas em seqüência em qualquer ordem. Elas não são executadas necessariamente na ordem em que são codificadas no conjunto de regras. Como cada ação deve ser concluída antes do início da próxima ação na seqüência, as ações não são executadas ao mesmo tempo.

Atributos

<onTimeOut> não possui atributos.

Contido em

<onTimeOut> está contido nos seguintes elementos:

- <sequenceRule>
- <thresholdRule>

Contém

<onTimeOut> contém o seguinte elemento:

Tabela 53. Elementos Contidos no Elemento <onTimeOut>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onTimeWindowComplete

O elemento <onTimeWindowComplete> é válido apenas para as regras de coleta, de computação, de duplicata e de cronômetro. Ele especifica a ação ou o conjunto de ações a serem executadas quando o espaço de tempo da regra expirar.

Detalhes

Se o conjunto de regras contiver várias regras que forem ativadas na mesma data e hora ou pelo mesmo evento e que possuírem o mesmo espaço de tempo, as seguintes ações para essas regras não serão executadas exatamente ao mesmo tempo:

- Ações de resposta da regra nos elementos <onTimeOut> e <onTimeWindowComplete>
- Ações de ciclo de vida nos elementos <onActivation> e <onDeactivation>

Essas ações são executadas em sequência em qualquer ordem. Elas não são executadas necessariamente na ordem em que são codificadas no conjunto de regras. Como cada ação deve ser concluída antes do início da próxima ação na sequência, as ações não são executadas ao mesmo tempo.

Atributos

<onTimeWindowComplete> não possui atributos.

Contido em

<onTimeWindowComplete> está contido nos seguintes elementos:

- <collectionRule>
- <computationRule>
- <duplicateRule>
- <timerRule>

Contém

<onTimeWindowComplete> contém o seguinte elemento:

Tabela 54. Elementos Contidos no Elemento <onTimeWindowComplete>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento onUnload

O elemento <onUnload> especifica a ação ou o conjunto de ações a serem executadas quando a regra é descarregada ou removida do mecanismo da Tecnologia ACT em execução. A ação <onUnload> é chamada imediatamente antes de a regra ser descarregada.

Atributos

<onUnload> não possui atributos.

Contido em

<onUnload> está contido no seguinte elemento:

- <lifeCycleActions>

Contém

<onUnload> contém o seguinte elemento:

Tabela 55. Elementos Contidos no Elemento <onUnload>

Elemento	Obrigatório ou opcional?
<action>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento ruleBlock

O elemento <ruleBlock> fornece a maneira para agrupar regras relacionadas e organizá-las em uma hierarquia.

Atributos

<ruleBlock> possui o seguinte atributo:

Tabela 56. Atributos do Elemento <ruleBlock>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica o bloco de regra. Esse identificador deve ser exclusivo no conjunto de regras ou no bloco de regra que contém esse bloco de regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim

Contido em

<ruleBlock> está contido nos seguintes elementos:

- <ruleSet>
- <ruleBlock>

Contém

<ruleBlock> contém os elementos a seguir.

Se estiverem codificados, os elementos <comment>, <import> e <variable> deverão ser codificados na ordem mostrada. Os outros elementos poderão ser codificados em qualquer ordem.

Tabela 57. Elementos Contidos no Elemento <ruleBlock>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<import>	Opcional. 0 ou mais ocorrências são permitidas.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<ruleBlock>	Opcional. 0 ou mais ocorrências são permitidas.
<collectionRule>	Opcional. 0 ou mais ocorrências são permitidas.

Tabela 57. Elementos Contidos no Elemento <ruleBlock> (continuação)

Elemento	Obrigatório ou opcional?
<computationRule>	Opcional. 0 ou mais ocorrências são permitidas.
<duplicateRule>	Opcional. 0 ou mais ocorrências são permitidas.
<filterRule>	Opcional. 0 ou mais ocorrências são permitidas.
<sequenceRule>	Opcional. 0 ou mais ocorrências são permitidas.
<thresholdRule>	Opcional. 0 ou mais ocorrências são permitidas.
<timerRule>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento ruleSet

O elemento <ruleSet> definido por act:ruleSet é o elemento raiz da linguagem de regra da Tecnologia ACT. Os outros elementos estão contidos neste elemento <ruleSet>.

Detalhes

Os elementos <ruleSet> definidos pelo esquema de linguagem da Tecnologia ACT (act:ruleSet) e pelo esquema do conjunto de regras base da Tecnologia ACT (br:ruleSet) são duplicatas. No entanto, ao criar um conjunto de regras, é necessário especificar o seguinte espaço de nomes no elemento <ruleSet>: act:ruleSet.

Atributos

<ruleSet> possui o seguinte atributo:

Tabela 58. Atributos do Elemento <ruleSet>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica o conjunto de regras. Esse identificador deve ser exclusivo. Não pode conter um ponto.	xsd:NMTOKEN	Sim

Contido em

Como <ruleSet> é o elemento raiz da linguagem de regras, ele não está contido em nenhum elemento.

Contém

<ruleSet> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 59. Elementos Contidos no Elemento <ruleSet>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<import>	Opcional. 0 ou mais ocorrências são permitidas.

Tabela 59. Elementos Contidos no Elemento <ruleSet> (continuação)

Elemento	Obrigatório ou opcional?
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<ruleBlock>	Opcional. 0 ou mais ocorrências são permitidas.

Elemento runUntilDeactivated

O elemento <runUntilDeactivated> especifica que o espaço de tempo continua aberto até que a regra seja desativada. Portanto, o espaço de tempo para essa regra inicia quando a regra começa a ser processada e não pára até que a regra seja desativada ou removida do conjunto de regras ou o mecanismo da Tecnologia ACT seja encerrada.

Detalhes

O comportamento específico de uma regra que inclui o elemento <runUntilDeactivated> depende do tipo de regra. Tabela 60 descreve o comportamento de cada tipo de regra em que o elemento <timeWindow> é válido e contém o elemento <runUntilDeactivated>.

Tabela 60. Comportamento da Regra quando <runUntilDeactivated> é Codificado

Tipo de regra	Comportamento da Regra quando <runUntilDeactivated> é Codificado
coleta	A regra de coleta aceita o primeiro evento que atender a seus critérios de seleção de eventos e continua a aceitar e a processar eventos até que a regra seja desativada, em cujo momento as ações definidas no elemento <onTimeWindowComplete> são executadas, seguidas imediatamente das ações definidas no elemento <onDeactivation>.
computação	A regra de computação aceita o primeiro evento que atender a seus critérios de seleção de eventos e continua a aceitar e a processar eventos até que a regra seja desativada, em cujo momento as ações definidas no elemento <onTimeWindowComplete> são executadas, seguidas imediatamente das ações definidas no elemento <onDeactivation>.
duplicata	A regra de duplicata aceita o primeiro evento que atender a seus critérios de seleção de eventos e continua a aceitar e a processar eventos até que a regra seja desativada, em cujo momento as ações definidas no elemento <onTimeWindowComplete> são executadas, seguidas imediatamente das ações definidas no elemento <onDeactivation>.
seqüência	<p>A regra de seqüência aceita o primeiro evento que atender a seus critérios de seleção de eventos e continua a aceitar e a processar eventos até que ocorra nas seguintes circunstâncias:</p> <ul style="list-style-type: none"> • O padrão de seqüência é detectado. Quando isso ocorre, as ações definidas no elemento <onDetection> são executadas e a regra retorna ao seu estado inicial. O processamento de eventos por essa regra começa novamente e esse processo pode ser repetido várias vezes até que a regra seja desativada. • A regra é desativada durante o processamento de eventos. Quando isso ocorre, as ações definidas no elemento <onTimeOut> são executadas, seguidas imediatamente das ações definidas no elemento <onDeactivation>.

Tabela 60. Comportamento da Regra quando `<runUntilDeactivated>` é Codificado (continuação)

Tipo de regra	Comportamento da Regra quando <code><runUntilDeactivated></code> é Codificado
limite	<p>A regra de limite aceita o primeiro evento que atender a seus critérios de seleção de eventos e continua a aceitar e a processar eventos até que ocorra nas seguintes circunstâncias:</p> <ul style="list-style-type: none"> • O padrão de limite é detectado. Quando isso ocorre, as ações definidas no elemento <code><onDetection></code> são executadas e a regra retorna ao seu estado inicial. O processamento de eventos por essa regra começa novamente e esse processo pode ser repetido várias vezes até que a regra seja desativada. • A regra é desativada durante o processamento de eventos. Quando isso ocorre, as ações definidas no elemento <code><onTimeOut></code> são executadas, seguidas imediatamente das ações definidas no elemento <code><onDeactivation></code>.
cronômetro	<p>Depois que a regra do cronômetro torna-se ativa, não acontece nada até que ela seja desativada, em cujo momento as ações definidas no elemento <code><onTimeWindowComplete></code> são executadas, seguidas imediatamente das ações definidas no elemento <code><onDeactivation></code>. O atributo de repetição no elemento <code><timerRule></code> é ignorado.</p>

Atributos

`<runUntilDeactivated>` não possui atributos.

Contido em

`<runUntilDeactivated>` está contido no seguinte elemento:

- `<timeWindow>`

Contém

`<runUntilDeactivated>` não contém elementos.

Elemento `sequenceRule`

O elemento `<sequenceRule>` define uma regra de acordo com o padrão de seqüência. A regra de seqüência é a única que permite vários seletores de eventos. Ela também requer um mínimo de dois seletores de eventos.

Atributos

`<sequenceRule>` possui os seguintes atributos:

Tabela 61. Atributos do Elemento `<sequenceRule>`

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim

Tabela 61. Atributos do Elemento <sequenceRule> (continuação)

Nome	Descrição	Tipo de dado	Obrigatório?
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não
arrivalOrder	Define se os eventos devem ocorrer na ordem em que os elementos <eventSelector> estão codificados na regra. Os valores válidos são: <ul style="list-style-type: none"> • inOrder (o valor padrão) • randomOrder 	xsd:string	Não

Se o valor do atributo arrivalOrder for randomOrder, a ordem de codificação dos elementos <eventSelector> é importante. Os elementos <eventSelector> com os critérios de seleção de eventos mais específicos devem ser codificados antes dos elementos <eventSelector> com critérios de seleção de eventos menos específicos. Caso contrário, a sequência não será detectada apropriadamente.

Por exemplo, suponha as seguintes circunstâncias:

- Os elementos <eventSelector> estão definidos.
- O primeiro elemento <eventSelector> está procurando o evento eventA.
- O segundo elemento <eventSelector> está procurando qualquer evento.
- O terceiro elemento <eventSelector> está procurando o evento eventB.
- Os seguintes eventos são apresentados ao sistema no espaço de tempo especificado: eventA, eventB, eventC.

O comportamento da regra é a que segue, e a sequência não será detectada apropriadamente:

1. O primeiro evento, eventA, é aceito pelo primeiro elemento <eventSelector>.
2. O segundo evento, eventB, é aceito pelo segundo elemento <eventSelector>.
3. O terceiro evento, eventC, é ignorado.

Suponha as seguintes circunstâncias, em que os elementos <eventSelector> estão codificados corretamente, com os critérios de seleção de eventos mais específicos precedendo os menos específicos:

- O primeiro elemento <eventSelector> está procurando o evento eventA.
- O segundo elemento <eventSelector> está procurando o evento eventB.
- O terceiro elemento <eventSelector> está procurando qualquer evento.

O comportamento de regra é a que segue e a sequência será detectada:

1. O primeiro evento, eventA, é aceito pelo primeiro elemento <eventSelector>.
2. O segundo evento, eventB, é aceito pelo segundo elemento <eventSelector>.
3. O terceiro evento, eventC, é aceito pelo terceiro elemento <eventSelector>.

Contido em

<sequenceRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<sequenceRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 62. Elementos Contidos no Elemento <sequenceRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<eventSelector>	2 ocorrências desse elemento são obrigatórias para a regra de seqüência. Ocorrências adicionais são permitidas.
<groupingKey>	Opcional. 0 ou 1 ocorrência é permitida.
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onDetection>	Opcional. 0 ou 1 ocorrência é permitida.
<onTimeOut>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Seqüência” na página 14

Uma regra de seqüência é definida pelo padrão de seqüência. Ela detecta se uma determinada seqüência de eventos chega em um intervalo de tempo. A seqüência pode ser ordenada ou aleatória. Uma regra de seqüência é uma regra com preservação de estado.

Elemento start

O elemento <start> define se uma regra será ativada em um determinado momento em uma determinada data ou se será ativado quando a regra for carregada pelo mecanismo da Tecnologia ACT.

Detalhes

Se o elemento <start> não estiver mesmo codificado, a hora de início padrão será a mesma definida pelo elemento <whenLoaded>.

Atributos

<start> não possui atributos.

Contido em

<start> está contido no seguinte elemento:

- <activationTime>

Contém

<start> contém os seguintes elementos:

Tabela 63. Elementos Contidos no Elemento <start>

Elemento	Obrigatório ou opcional?
<dateTime>	1 desses elementos é obrigatório e apenas 1 ocorrência do elemento escolhido é permitida.
<whenLoaded>	
<inactiveWhenLoaded>	

Elemento stop

O elemento <stop> define se uma regra será desativada em um determinado momento em uma determinada data, depois de uma determinada duração, ou nunca em um determinado momento.

Detalhes

Se o elemento <stop> não estiver mesmo codificado, a hora de parada padrão será a mesma definida pelo elemento <never>.

Atributos

O elemento <stop> não possui atributos.

Contido em

O elemento <stop> está contido no seguinte elemento:

- <activationTime>

Contém

O elemento <stop> contém os seguintes elementos:

Tabela 64. Elementos Contidos no Elemento <stop>

Elemento	Obrigatório ou opcional?
<dateTime>	1 desses elementos é obrigatório e apenas 1 ocorrência do elemento escolhido é permitida.
<never>	
<after>	

Elemento stopAfter

O elemento <stopAfter> especifica a duração de tempo que uma instância da regra, conforme definido pelo elemento <groupingKey>, permanecerá ativa depois que se tornar ativa. Depois desse tempo, a instância da regra deverá ser desativada.

Atributos

O elemento <stopAfter> possui os seguintes atributos:

Tabela 65. Atributos do Elemento <stopAfter>

Nome	Descrição	Tipo de dado	Obrigatório?
duration	Especifica o período da duração. O tipo de dado desse atributo depende do valor do atributo de unidade.	<ul style="list-style-type: none">Se o valor do atributo de unidade for ISO-8601, o tipo de dado será xsd:duration.Se o valor do atributo de unidade for milliseconds, o tipo de dado será xsd:positiveInteger.	Sim
unit	Especifica a unidade de tempo a ser utilizada. Os valores válidos para esse atributo são: <ul style="list-style-type: none">ISO-8601milliseconds	xsd:string	Sim

O uso do ISO 8601 padrão para duração de tempo

A codificação ISO-8601 como o valor para o atributo de unidade indica que o valor do atributo de duração é codificado de acordo com o padrão ISO 8601 para especificar uma duração de tempo como uma cadeia. A especificação de tipo de dado do esquema XML padrão utiliza ISO 8601 para fornecer um tipo de dado chamado duration. Esse tipo de dado é descrito em detalhes em <http://www.w3.org/TR/xmlschema-2/#duration>.

O formato do tipo de dado duration no esquema XML padrão é a seguinte cadeia:

PnYnMnDTnHnMnS

- P é o caractere que sempre começa a cadeia.
- nY representa o número de anos. Um ano é o mesmo que 365 dias. Portanto, a codificação 1Y é o mesmo que a codificação 365D.
- nM representa o número de meses. Um mês é o mesmo que 30 dias. Portanto, a codificação 1M é o mesmo que a codificação 30D.
- nD representa o número de dias.
- T é um separador que separa as unidades de dias (anos, meses e dias) das unidades de tempo (horas, minutos e segundos). As unidades de tempo sempre seguem T.
- nH representa o número de horas.
- nM representa o número de minutos.
- nS representa o número de segundos.

A seguir, exemplos do formato:

- P5DT12H é 5,5 dias.
- PT59M59S é 59 minutos e 59 segundos.
- P1M é 1 mês.

Contido em

<stopAfter> está contido no elemento <activateOnEvent>, mas apenas quando <activateOnEvent> está codificado no elemento <activationByGroupingKey>.

Contém

<stopAfter> não contém elementos.

Elemento thresholdRule

O elemento <thresholdRule> define uma regra de acordo com o padrão de limite.

Atributos

<thresholdRule> possui os seguintes atributos:

Tabela 66. Atributos do Elemento <thresholdRule>

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	xsd:NMTOKEN	Sim
processOnlyForwardedEvents	Determina se a regra recebe todos os eventos ou apenas aqueles redirecionados a partir de outras regras. O valor padrão é false, indicando que a regra recebe todos os eventos, incluindo aqueles redirecionados a partir de outras regras.	xsd:boolean	Não

Contido em

<thresholdRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<thresholdRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 67. Elementos Contidos no Elemento <thresholdRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.

Tabela 67. Elementos Contidos no Elemento <thresholdRule> (continuação)

Elemento	Obrigatório ou opcional?
<eventSelector>	Opcional. 0 ou 1 ocorrência é permitida.
<groupingKey>	Opcional. 0 ou 1 ocorrência é permitida.
<booleanThreshold>	1 desses elementos é obrigatório e apenas 1 ocorrência do elemento escolhido é permitida.
<computedThreshold>	
<eventCountThreshold>	
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onDetection>	Opcional. 0 ou 1 ocorrência é permitida.
<onTimeOut>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Limite” na página 17

Uma regra de limite é definida pelo padrão de limite. Ela coleta um grupo de eventos selecionados em um intervalo de tempo e determina, após a recepção de cada evento, se uma condição de limite foi atendida. É uma regra com preservação de estado.

Elemento timeInterval

O elemento <timeInterval> especifica a duração do espaço de tempo.

Atributos

<timeInterval> possui os seguintes atributos:

Tabela 68. Atributos do Elemento <timeInterval>

Nome	Descrição	Tipo de dado	Obrigatório?
duration	Especifica o período da duração. O tipo de dado desse atributo depende do valor do atributo de unidade.	<ul style="list-style-type: none"> Se o valor do atributo de unidade for ISO-8601, o tipo de dado será xsd:duration. Se o valor do atributo de unidade for milliseconds, o tipo de dado será xsd:positiveInteger. 	Sim
unit	Especifica a unidade de tempo a ser utilizada. Os valores válidos para esse atributo são: <ul style="list-style-type: none"> ISO-8601 milliseconds 	xsd:string	Sim

O uso do ISO 8601 padrão para duração de tempo

A codificação ISO-8601 como o valor para o atributo de unidade indica que o valor do atributo de duração é codificado de acordo com o padrão ISO 8601 para especificar uma duração de tempo como uma cadeia. A especificação de tipo de dado do esquema XML padrão utiliza ISO 8601 para fornecer um tipo de dado chamado duration. Esse tipo de dado é descrito em detalhes em <http://www.w3.org/TR/xmlschema-2/#duration>.

O formato do tipo de dado `duration` no esquema XML padrão é a seguinte cadeia:

`PnYnMnDTnHnMnS`

- `P` é o caractere que sempre começa a cadeia.
- `nY` representa o número de anos. Um ano é o mesmo que 365 dias. Portanto, a codificação `1Y` é o mesmo que a codificação `365D`.
- `nM` representa o número de meses. Um mês é o mesmo que 30 dias. Portanto, a codificação `1M` é o mesmo que a codificação `30D`.
- `nD` representa o número de dias.
- `T` é um separador que separa as unidades de dias (anos, meses e dias) das unidades de tempo (horas, minutos e segundos). As unidades de tempo sempre seguem `T`.
- `nH` representa o número de horas.
- `nM` representa o número de minutos.
- `nS` representa o número de segundos.

A seguir, exemplos do formato:

- `P5DT12H` é 5,5 dias.
- `PT59M59S` é 59 minutos e 59 segundos.
- `P1M` é 1 mês.

Contido em

`<timeInterval>` está contido no seguinte elemento:

- `<timeWindow>`

Contém

`<timeInterval>` não contém elementos.

Elemento `timerRule`

O elemento `<timerRule>` define uma regra de acordo com o padrão de cronômetro.

Atributos

`<timerRule>` possui os seguintes atributos:

Tabela 69. Atributos do Elemento `<timerRule>`

Nome	Descrição	Tipo de dado	Obrigatório?
<code>name</code>	Identifica a regra. Esse identificador deve ser exclusivo no bloco de regra que contém essa regra. Não pode conter um ponto.	<code>xsd:NMTOKEN</code>	Sim
<code>processOnlyForwardedEvents</code>	Esse atributo é ignorado porque a regra do cronômetro não processa eventos.	<code>xsd:boolean</code>	Não

Tabela 69. Atributos do Elemento <timerRule> (continuação)

Nome	Descrição	Tipo de dado	Obrigatório?
repeat	<p>Define se a regra do cronômetro será executada repetidamente até que seja desativada. Os valores válidos são:</p> <ul style="list-style-type: none"> • true (o valor padrão) • false <p>Se o valor for definido como false, a regra será executada por meio de seu intervalo de tempo apenas uma vez, executará a ação de resposta da regra quando o espaço de tempo respectivo estiver concluído e será parada.</p> <p>Se o elemento <timeWindow> da regra do cronômetro contiver o elemento <runUntilDeactivated>, o atributo repeat será ignorado.</p>	xsd:boolean	Não

Contido em

<timerRule> está contido no seguinte elemento:

- <ruleBlock>

Contém

<timerRule> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 70. Elementos Contidos no Elemento <timerRule>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<variable>	Opcional. 0 ou mais ocorrências são permitidas.
<activationInterval>	Opcional. 0 ou 1 ocorrência é permitida.
<lifeCycleActions>	Opcional. 0 ou 1 ocorrência é permitida.
<timeWindow>	Obrigatório. Apenas 1 ocorrência é permitida.
<onTimeWindowComplete>	Opcional. 0 ou 1 ocorrência é permitida.

Conceitos relacionados

“Padrão de Cronômetro” na página 19

Uma regra de cronômetro é definida pelo padrão de cronômetro. Ele inicia

ações em intervalos regulares. É uma regra com preservação de estado. Embora uma regra do cronômetro não processe eventos, ela pode ser ativada ou desativada por um evento.

Elemento timeWindow

O elemento <timeWindow> contém elementos que definem o intervalo de tempo durante o qual a regra está sendo processada.

Detalhes

Por exemplo, o espaço de tempo para uma regra de duplicata define quanto tempo a regra deve procurar eventos que sejam duplicatas do primeiro evento que for aceito. Se o espaço de tempo for de 30 segundos, a regra de duplicata processará todos os eventos de duplicata ocorridos nos 30 segundos do primeiro evento que for aceito.

Atributos

<timeWindow> não possui atributos.

Contido em

<timeWindow> está contido nos seguintes elementos:

- <collectionRule>
- <computationRule>
- <duplicateRule>
- <sequenceRule>
- <thresholdRule>
- <timerRule>

Contém

<timeWindow> contém os seguintes elementos:

Tabela 71. Elementos Contidos no Elemento <timeWindow>

Elemento	Obrigatório ou opcional?
<timeInterval>	1 desses elementos é obrigatório e apenas 1 ocorrência do elemento escolhido é permitida.
<runUntilDeactivated>	

Elemento variable

O elemento <variable> define uma variável e contém informações em uma forma que pode ser referenciada por expressões. Uma variável pode ser definida no nível do conjunto de regras, de um bloco de regra ou de uma regra.

Detalhes

variável do conjunto de regras

Aplica-se globalmente ao conjunto de regras e pode ser referenciada por qualquer expressão nesse conjunto de regras.

variável do bloco de regra

Aplica-se apenas no bloco de regra (e em qualquer bloco de regra contido) e pode ser referenciada por qualquer expressão nesse bloco de regra.

variável da regra

Aplica-se apenas a expressões nessa regra.

As variáveis podem ter o mesmo nome em níveis diferentes na hierarquia da regra. Quando uma variável é acessada, é utilizada sua definição mais local. Por exemplo, se uma variável for definida no nível do conjunto de regras, nível do bloco de regra e nível de regra com o mesmo nome, a definição da variável no nível de regra será utilizada pelas expressões dessa regra.

Quando as variáveis forem definidas no nível do conjunto de regras ou do bloco de regra, várias regras obterão e definirão essas variáveis em momentos diferentes. Portanto, para assegurar que os valores das variáveis sejam mantidos corretamente, preste atenção em como codificará as interações entre as variáveis no conjunto de regras.

Se a variável for definida no nível do conjunto de regras ou do bloco de regra, ela não será reinicializada depois que o padrão de regra for correspondido.

Em uma das condições a seguir, utilize o bloqueio na obtenção e na configuração do conjunto de regras e das variáveis do bloco de regra para evitar a configuração incorreta dos valores de variáveis:

- Se a regra do cronômetro obtiver ou configurar uma variável durante uma ação `<onTimeout>`
- Se o aplicativo no qual o mecanismo da Tecnologia ACT estiver incorporado, for multithreadado

Se uma regra for definida com uma chave de agrupamento, as variáveis de regra definidas pelo elemento `<variable>` não serão válidas nas ações do ciclo de vida ou em um elemento `<filteringPredicate>` contido em um elemento `<activateOnEvent>` ou `<deactivateOnEvent>` contido em um elemento `<activationInterval>`. Isso ocorre porque, nesse caso, as variáveis da regra são aplicadas somente a uma instância da regra e as instâncias de regras não existem na hora em que essas expressões são executadas.

Atributos

`<variable>` possui os seguintes atributos:

Tabela 72. Atributos do Elemento `<variable>`

Nome	Descrição	Tipo de dado	Obrigatório?
name	Identifica uma variável específica. Uma variável é referenciada por seu nome.	xsd:NMTOKEN	Sim
dataType	Identifica o tipo de informações que uma variável contém. Deve ser um tipo de dado completo, como <code>java.lang.String</code> .	xsd:NMTOKEN	Sim

Restrições de Nomes para Variáveis

Os nomes de variáveis possuem certas restrições. Portanto, o valor do atributo name no elemento <variable> possui as seguintes restrições:

- Pode incluir apenas os seguintes caracteres:
 - Letras latinas A-Z em ASCII maiúsculas. A representação em Unicode é \u0041-\u005a.
 - Letras latinas a-z em ASCII minúsculas. A representação em Unicode é \u0061-\u007a.
 - O sublinhado em ASCII (_). A representação em Unicode é \u005f.
 - O sinal de dólar (\$). A representação em Unicode é \u0024.
 - Os dígitos ASCII de 0 a 9. A representação em Unicode é \u0030-\u0039.
- Não pode ser nulo.
- Não pode ser uma cadeia vazia.
- Não pode conter espaços em branco.
- Não pode conter um ponto.
- Não pode iniciar com act_ em nenhuma forma (letras maiúsculas, minúsculas ou misturadas).

Contido em

<variable> está contido nos seguintes elementos:

- <ruleSet>
- <ruleBlock>
- <collectionRule>
- <computationRule>
- <duplicateRule>
- <filterRule>
- <sequenceRule>
- <thresholdRule>
- <timerRule>

Contém

<variable> contém os elementos a seguir.

Os elementos devem ser codificados na ordem mostrada. Se um elemento for opcional, ele não precisará ser codificado, mas todos os elementos que forem codificados deverão seguir a ordem correta.

Tabela 73. Elementos Contidos no Elemento <variable>

Elemento	Obrigatório ou opcional?
<comment>	Opcional. 0 ou 1 ocorrência é permitida.
<varInitializer>	Obrigatório. 1 ocorrência é permitida.

Conceitos relacionados

“Variáveis” na página 24

Na linguagem de regras, determinadas variáveis são utilizadas para armazenar informações relacionadas a eventos nas diferentes ocorrências ou regras do evento. Essas informações relacionadas a eventos poderão, então, ser acessadas

a partir de expressões nas regras. Alguns tipos de variáveis são definidos pelo autor de regras e outros são fornecidos pela Tecnologia ACT. Alguns tipos podem ser acessados diretamente de uma expressão e outros podem ser acessados apenas por meio de métodos fornecidos pela Tecnologia ACT.

Elemento `varInitializer`

O elemento `<varInitializer>` contém uma expressão que fornece o valor inicial para a variável definida no elemento `<variable>` associado.

Detalhes

Como a variável pode ser de qualquer tipo, o código de expressão pode retornar um objeto de matriz ou qualquer outro objeto complexo específico da implementação a ser armazenado pelo mecanismo da Tecnologia ACT.

Consulte “Variáveis” na página 24 para obter informações sobre as variáveis que podem ser utilizadas em expressões. O uso de determinadas variáveis depende do contexto da expressão.

Atributos

`<varInitializer>` possui o seguinte atributo:

Tabela 74. Atributos do Elemento `<varInitializer>`

Nome	Descrição	Tipo de dado	Obrigatório?
<code>expressionLanguage</code>	Identifica a linguagem de programação na qual a expressão é gravada. Como a linguagem de programação Java é a única linguagem de expressão suportada, o único valor válido para esse atributo é <code>java</code> .	<code>xsd:NMTOKEN</code>	Sim

Contido em

`<varInitializer>` está contido no seguinte elemento:

- `<variable>`

Contém

`<varInitializer>` não contém elementos.

Conceitos relacionados

“Expressões” na página 20

Uma expressão é código que contém lógica personalizada que pode ser incluída em uma regra. As expressões também podem acessar código externo para o mecanismo da Tecnologia ACT. Na linguagem de regra, as expressões são válidas somente nos contextos específicos ou nos elementos de linguagem da regra.

Elemento `whenLoaded`

O elemento `<whenLoaded>` especifica que uma regra será ativada quando for carregada pelo mecanismo da Tecnologia ACT.

Atributos

<whenLoaded> não possui atributos.

Contido em

<whenLoaded> está contido no seguinte elemento:

- <start>

Contém

<whenLoaded> não contém elementos.

Capítulo 6. Glossário

Este glossário contém os termos e as definições de conceitos importantes na Tecnologia ACT.

ACT Consulte Tecnologia ACT.

ação Uma expressão executada como parte de uma resposta da regra ou quando uma regra é carregada, descarregada, ativada ou desativada.

Tecnologia ACT

Uma tecnologia da IBM que fornece a correlação de eventos por meio de regras.

compilador da Tecnologia ACT

O componente Tecnologia ACT que analisa um conjunto de regras e qualquer código contido nele para gerar as estruturas interna de dados requeridas pelo mecanismo da Tecnologia ACT.

mecanismo da Tecnologia ACT

O componente Tecnologia ACT que processa eventos de acordo com a saída do compilador da Tecnologia ACT.

construtor de regras da Tecnologia ACT

Uma GUI para gravar regras de correlação na linguagem de regras da Tecnologia ACT.

linguagem de regras da Tecnologia ACT

Uma linguagem baseada em XML para gravar regras para correlacionar eventos. Essas regras poderão então ser implementadas em ambientes de tempo de execução da Tecnologia ACT.

ambiente de tempo de execução da Tecnologia ACT

Um aplicativo no qual o mecanismo da Tecnologia ACT é incorporado, com ou sem o compilador.

padrão de coleta

Um padrão de regra que define uma regra para coletar um grupo de eventos selecionados em um intervalo de tempo. Uma regra definida pelo padrão de coleta é uma regra com preservação de estado.

padrão de computação

Um padrão de regra que define uma regra para aplicar um cálculo (por meio de uma expressão) a eventos coletados conforme cada evento é recebido em um intervalo de tempo. Uma regra definida pelo padrão de computação é uma regra sem preservação de estado.

domínio

A categoria à qual um grupo de regras é aplicado com base em sua função. Por exemplo, um domínio pode representar uma área geográfica específica, a disciplina de gerenciamento de TI (como detecção de segurança ou correlação de eventos da rede) ou a organização de negócios (como uma empresa específica ou uma divisão com uma empresa).

padrão de duplicata

Um padrão de regra que define uma regra para contar o segundo e os subseqüentes eventos aceitos no intervalo de tempo especificado, mas para ignorar o processamento do conjunto de regras deles. Uma regra definida pelo padrão de duplicata é uma regra com preservação de estado.

provedor de eventos

Qualquer software que gere eventos processados pela Tecnologia ACT.

seletor de eventos

Os critérios de seleção de eventos. Esses critérios determinam quais eventos são aceitos para processamento por uma regra. O seletor de eventos inclui o tipo de evento e o predicado de filtragem.

expressão

Código que contém lógica personalizada que pode ser incluída em uma regra. Os autores de regras podem utilizar as expressões com fins diferentes, como a inicialização de variáveis, a definição de critérios de seleção de eventos ou a especificação de ações de resposta da regra e de ciclo de vida.

linguagem de expressão

A linguagem de programação na qual uma expressão é gravada.

evento externo

Um evento que o mecanismo da Tecnologia ACT recebe de uma origem externa a ele.

objeto externo

Um objeto que um aplicativo cria para se comunicar com as expressões.

predicado de filtragem

Uma expressão que define a condição sob a qual um evento é aceito para processamento por uma regra. O predicado de filtragem é uma parte de um seletor de eventos. Um predicado de filtragem retorna um valor booleano.

padrão de filtragem

Um padrão de regra que define uma regra para executar uma determinada ação ao aceitar um evento. Uma regra definida pelo padrão de filtragem age apenas em um único evento e é, portanto, uma regra sem preservação de estado.

chave de agrupamento

Um método para direcionar uma regra para criar uma instância de regra separada (ou uma cópia de si mesma) para cada grupo de eventos que compartilham características comuns.

importação

Uma maneira específica da linguagem de programação tornar código externo acessível a expressões.

evento interno

Um evento criado por uma regra que está em execução no mecanismo da Tecnologia ACT. Esse evento pode ser redirecionado para outras regras.

ação de ciclo de vida

Uma expressão executada quando uma regra é carregada, descarregada, ativada ou desativada.

nó

Um objeto em uma hierarquia de regra que pode ser incluído, removido ou substituído individual e independentemente em um conjunto de regras. Especificamente, estes objetos são nós:

- Regras
- Blocos de regras
- Variáveis de blocos de regras
- Variáveis de conjuntos de regras

Como um objeto não pode ser operado individual e independentemente abaixo do nível de regra, uma variável de regra não é um nó.

predicado

Consulte predicado de filtragem.

resposta

Consulte resposta da regra.

regra A unidade de correlação utilizada para reconhecer relacionamentos entre os eventos e para executar respostas apropriadas da regra. Uma regra é uma implementação de um de sete padrões de regras e é organizada, de acordo com sua função, em um bloco de regra que faz parte de um conjunto de regras. Uma regra aceitará um evento para processamento se o evento atender aos critérios de seleção de eventos.

bloco de regra

A organizational unit para agrupamento de regras por função em domínios no conjunto de regras. Um bloco de regra não pode conter apenas regras, mas também outros blocos de regras.

instância de regra

No contexto da chave de agrupamento, uma cópia de uma regra.

padrão de regra

A representação de uma situação de correlação de eventos (como uma condição de limite ou uma detecção de eventos de duplicata). A linguagem de regra da Tecnologia ACT inclui os seguintes padrões de regras: coleta, computação, duplicata, filtragem, seqüência, limite e cronômetro. O padrão de uma regra é correspondido quando a situação definida pela regra ocorre. Quando o padrão é correspondido, a regra conclui seu processamento executando as ações apropriadas de resposta da regra. Enquanto uma regra está ativa, o padrão de regra pode ser correspondido várias vezes.

resposta da regra

Uma expressão executada quando o mecanismo da Tecnologia ACT reconhece que uma condição de regra foi atendida. Uma resposta de regra consiste em uma ou mais ações.

ação de resposta da regra

Consulte ação.

conjunto de regras

A unidade de execução da regra para a linguagem de regra da Tecnologia ACT. O conjunto de regras contém as regras, organizadas em blocos de regras, a serem executadas pelo mecanismo da Tecnologia ACT. O mecanismo age apenas em um único conjunto de regras em uma determinada hora.

padrão de seqüência

Um padrão de regra que define uma regra para detectar a presença ou a ausência de uma determinada seqüência de eventos em um intervalo de tempo. A seqüência pode ser ordenada ou aleatória. Uma regra definida pelo padrão de seqüência é uma regra com preservação de estado.

fragmento

Uma seleção de código fonte.

regra com preservação de estado

Uma regra que retém informações de estado, que são informações sobre as características de uma instância de regra, com o fim de agir em uma coleta

de eventos em um período de tempo. As regras definidas por um dos padrões de regras a seguir são regras com preservação de estado: coleta, computação, duplicata, seqüência, limite ou cronômetro.

regra sem preservação de estado

Uma regra que não retém informações de estado e, portanto, podem agir apenas em um evento de cada vez. Uma regra definida pelo padrão de filtragem é uma regra sem preservação de estado.

padrão de limite

Um padrão de regra que define uma regra para coletar um grupo de eventos selecionados em um intervalo de tempo e para determinar, após a recepção de cada evento, se uma condição de limite foi atendida. Uma regra definida pelo padrão de limite é uma regra com preservação de estado.

padrão de cronômetro

Um padrão de regra que define uma regra para iniciar ações em intervalos regulares. Uma regra definida pelo padrão de cronômetro é uma regra com preservação de estado. Embora uma regra do cronômetro não processe eventos, ela pode ser ativada ou desativada por um evento.

Apêndice. Avisos

Estas informações foram desenvolvidas para produtos e serviços oferecidos nos EUA. A IBM pode não oferecer os produtos, serviços ou recursos discutidos neste documento em outros países. Consulte um representante IBM local para obter informações sobre produtos e serviços disponíveis atualmente em sua área. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM, poderá ser utilizado em substituição a este produto, programa ou serviço. Entretanto, a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM são de responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não concede ao Cliente nenhuma licença sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP: 22290-240

Para pedidos de licenças com relação a informações sobre DBCS (Conjunto de Caracteres de Byte Duplo), entre em contato com o Departamento de Propriedade Intelectual da IBM em seu país ou envie pedidos, por escrito, para:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

O parágrafo a seguir não se aplica ao Reino Unido ou a qualquer outro país em que tais disposições não estejam de acordo com a legislação local:

A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO NO ESTADO EM QUE SE ENCONTRA, SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO SE LIMITANDO ÀS GARANTIAS IMPLÍCITAS DE MERCADO OU DE ADEQUAÇÃO A UM DETERMINADO PROPÓSITO.

Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações; portanto, esta declaração pode não se aplicar ao Cliente.

Estas informações podem conter imprecisões técnicas ou erros tipográficos. Periodicamente, são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar os produtos e/ou programas descritos nesta publicação, sem aviso prévio.

Referências nestas informações a Web sites não-IBM são fornecidas apenas por conveniência e não representam de forma alguma um endosso a esses Web sites. Os materiais contidos nesses Web sites não fazem parte dos materiais desse produto IBM e a utilização desses Web sites é de inteira responsabilidade do Cliente.

A IBM pode utilizar ou distribuir as informações fornecidas da forma que julgar apropriada sem incorrer em qualquer obrigação para com o Cliente.

Licenciados deste programa que desejam obter informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil
Av. Pasteur, 138-146
Botafogo
Rio de Janeiro, RJ
CEP: 22290-240

Tais informações podem estar disponíveis, sujeitas aos termos e condições apropriados, incluindo, em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, do Contrato de Licença do Programa Internacional IBM ou de qualquer contrato equivalente celebrado entre as partes.

Marcas Registradas

DB2, IBM, o logotipo IBM, Tivoli, o logotipo da Tivoli, Tivoli Enterprise Console e WebSphere são marcas registradas e ou comerciais da International Business Machines Corporation, nos EUA e/ou em outros países.

Java e todas as marcas registradas e logotipos baseados em Java são marcas registradas ou comerciais da Sun Microsystems, Inc., nos Estados Unidos e/ou em outros países.

Outros nomes de empresas, produtos e serviços podem ser marcas registradas ou de serviço de terceiros.



Impresso em Brazil