





Nota

Prima di utilizzare le seguenti informazioni e il prodotto supportato, consultare le informazioni contenute nel paragrafo "Informazioni particolari", a pagina 125.

Note informative

Queste informazioni descrivono IBM Active Correlation Technology ed il suo ruolo nell'elaborazione complessa degli eventi. Il linguaggio di Active Correlation Technology si basa su-XML per la scrittura delle regole per correlare gli eventi. Queste informazioni sono indirizzate agli scrittori di regole che devono comprendere come scrivere le regole per correlare gli eventi nella propria azienda.

Indice

Note informative	iii
----------------------------	-----

Parte 1. Guida al programma di scrittura delle regole 1

Capitolo 1. Introduzione 3

Capitolo 2. Panoramica sul linguaggio delle regole 5

Anatomia di una regola	5
Ciclo di vita di una regola	7
Organizzazione delle regole	9
Schemi di regola	11
Schema di raccolta	11
Schema di calcolo	12
Schema di duplicato	12
Schema di filtro	13
Schema di sequenza	14
Schema di soglia	16
Schema timer	18
Aspetti comuni e univoci di schemi di regola differenti	19
Espressioni	20
Importazione e accesso ai moduli e oggetti esterni	21
Inizializzazione e accesso alle variabili	22
Accesso alle informazioni correlate all'evento	23
Procedure ottimali per le espressioni di codifica	23
Variabili	24
Tipi di dati per le variabili di Active Correlation Technology	26
Contesti di espressione in cui sono valide le variabili	26
Variabile act_event	27
Variabile act_eventCount	28
Variabile act_eventList	29
Variabile act_lib	29
Variabile act_location	31
Variabile act_nodeName	32
Variabile act_threshold	33
Flusso di eventi tramite una serie di regole	33

Capitolo 3. Panoramica sulla scrittura delle regole 35

Pianificazione per la correlazione eventi	35
Designazione delle regole da correlare agli eventi	37
Introduzione al programma di creazione di regole	38
Impostazione di prospettiva in Eclipse Workbench	38
Impostazione delle preferenze	39
Creazione di un progetto per la memorizzazione di un file di una serie di regole	39
Creazione di una serie di regole	40
Creazione di un blocco di regole	40

Creazione di una regola	41
Convalida di una serie di regole	41
Compilazione di una serie di regole	42
Aggiornamento di una serie di regole	42
Inclusione di frammenti nelle espressioni all'interno delle regole	42

Parte 2. Riferimenti per il programma di scrittura delle regole. 45

Capitolo 4. Riepilogo sull'organizzazione della serie di regole 47

Riepilogo della serie di regole	47
Riepilogo del blocco di regole	47
Riepilogo di regola Collection	48
Riepilogo della regola computation	50
Riepilogo regola di duplicato	51
Riepilogo di regola di filtro	52
Riepilogo della regola di sequenza	53
Riepilogo della regola di soglia (Threshold).	55
Riepilogo di regola Timer	56

Capitolo 5. Panoramica sugli elementi di linguaggio 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	75
Elemento collectionRule	75
Elemento comment	77
Elemento computationRule	77
Elemento computedThreshold	78
Elemento computedValue	80
Elemento computeFunction	81
Elemento dateTime	82
Elemento deactivateOnEvent	84
Elemento duplicateRule	85
Elemento eventAttribute	86
Elemento eventCountThreshold	86
Elemento eventSelector	89
Elemento eventType	90
Elemento filteringPredicate	91
Elemento filterRule	92
Elemento groupingKey	93
Elemento import	96
Elemento inactiveWhenLoaded	97
Elemento lifeCycleActions	97
elemento never	98
Elemento onActivation	98

Elemento onDeactivation	99
Elemento onDetection	100
Elemento onLoad	100
Elemento onNextEvent	101
Elemento onTimeOut	102
Elemento onTimeWindowComplete	103
Elemento onUnload	103
Elemento ruleBlock	104
Elemento ruleSet	105
Elemento runUntilDeactivated	106
Elemento sequenceRule	107
Elemento start	109
Elemento stop	110
Elemento stopAfter	110

Elemento thresholdRule	112
Elemento timeInterval	113
Elemento timerRule	114
Elemento timeWindow	116
Elemento variable	116
Elemento varInitializer	118
Elemento whenLoaded	119

Capitolo 6. Glossario 121

Appendice. Informazioni particolari 125

Marchi	126
------------------	-----

Parte 1. Guida al programma di scrittura delle regole

Capitolo 1. Introduzione

Questa introduzione descrive in breve l'elaborazione CEP (complex event processing) e fornisce una panoramica dell'Active Correlation Technology e del suo ruolo in questo tipo di elaborazione.

L'ambiente aziendale moderno

Oggi, le organizzazioni commerciali e pubbliche dipendono dall'elaborazione di informazioni elettroniche tramite reti di computer e specialmente tramite Internet. Con l'arrivo di nuove tecnologie, quali l'elaborazione a griglie, le organizzazioni utilizzano applicazioni, in qualsiasi momento e luogo del mondo, aventi finalità di importanza critica. I processi, le attività e l'infrastruttura aziendali e quindi tutta la società dipendono dai livelli di IT (Information Technology) delle organizzazioni.

Le organizzazioni devono conoscere cosa succede nelle proprie aziende in qualsiasi momento. Ad esempio esse devono conoscere se sono disponibili e funzionano in modo appropriato le applicazioni aventi finalità di importanza critica e sapere come rilevare ed evitare possibili crisi nei processi, nelle attività o nell'infrastruttura aziendali. Se si verifica una crisi, bisogna immediatamente capire il problema, come risolverlo e cosa lo ha causato.

Il significato della maggior parte degli eventi relativi ai processi, alle attività e alla infrastruttura aziendali non viene mai riconosciuto o capito poiché la quantità di informazioni è troppo estesa e troppo difficile da elaborare in quanto presente su parti singole e non correlate. Tuttavia se gli eventi vengono aggregati e correlati in modo da rendere comprensibili le loro relazioni, essi possono instradare molte informazioni.

Lo scopo del CEP (complex event processing) è quello di ottenere migliori informazioni sugli eventi in tempo reale.

CEP (complex event processing)

Un *evento* è semplicemente una notifica su qualcosa che si è verificato.

CPE (complex event processing) è una derivazione degli eventi di alto livello dall'analisi, dalla correlazione e dalla sintesi di eventi di basso livello, nei sistemi basati sugli eventi. Questi eventi di alto livello, chiamati eventi complessi, sono adatti per rendere note le opportunità o i problemi aziendali, in modo facilmente intelligibile, o per attivare i processi automatizzati. Le organizzazioni possono quindi operare in modo più efficiente usufruendo di anticipazioni su potenziali opportunità o problemi e di una migliore comprensione delle cause principali che modificano le condizioni nei propri processi, attività e infrastruttura aziendali.

La *correlazione eventi* è il processo di definizione e rilevamento degli schemi nei flussi di evento in tempo reale e dell'implementazione di azioni in risposta agli eventi correlati. Essa è utilizzata per identificare un problema in base ai sintomi rilevati. Gli eventi possono essere correlati per causa, tempo, appartenenza o combinazioni. La correlazione eventi è parte integrale dell'elaborazione dell'evento.

Active Correlation Technology

Active Correlation Technology utilizza le regole per rilevare gli schemi nei flussi di eventi in tempo reale. Questa tecnologia si basa sul fatto che in molti casi le azioni di risposta non vengono attivate da un singolo evento di basso livello ma piuttosto da una composizione di eventi che si verificano in momenti diversi e in contesti differenti. Active Correlation Technology fa uso delle relazioni tra eventi per apprendere le opportunità e problemi aziendali. Ad esempio in base a ciò che si ottiene tramite la correlazione di eventi in tempo reale, un'organizzazione può intraprendere i seguenti tipi di azioni:

- Offrire spedizioni scontate per alcuni o tutti i clienti durante una vendita festiva.
- Nei successivi 30 giorni, calcolare il costo della spedizione in base al vettore della spedizione e la quantità dell'ordine.
- Fornire ai clienti che acquistano beni per più di 500 euro tra il primo luglio 2006 e il 31 dicembre 2006 un buono omaggio del valore di 25 euro.
- Notificare ad un amministratore che l'elaborazione dell'ordine non è stata completata entro le 36 ore.
- Notificare ad un amministratore che sono stati rilevati più di quattro tentativi di collegamento allo stesso computer in 30 secondi.

Active Correlation Technology consiste di quattro elementi primari:

Linguaggio delle regole dell'Active Correlation Technology

Un linguaggio basato su XML per la scrittura dei regole per correlare gli eventi. Queste regole possono essere distribuite sugli ambienti di esecuzione dell'Active Correlation Technology.

Engine dell'Active Correlation Technology

Il componente dell'Active Correlation Technology che elabora gli eventi in base all'output del compilatore dell'Active Correlation Technology.

Programma di creazione regole dell'Active Correlation Technology

Una GUI per scrivere le regole di correlazione nel linguaggio regole dell'Active Correlation Technology.

L'ambiente di esecuzione dell'Active Correlation Technology è un'applicazione in cui è integrato l'engine dell'Active Correlation Technology.

Capitolo 2. Panoramica sul linguaggio delle regole

Questa panoramica descrive i concetti chiave del linguaggio di regole dell'Active Correlation Technology.

Uno schema di regola è una rappresentazione di una condizione di correlazione eventi (come una condizione di soglia o un rilevamento di evento duplicato). Il linguaggio di regole di Active Correlation Technology include sette schemi di regole che sono stati provati per rappresentare la maggior parte delle situazioni di correlazione eventi a cui i clienti IBM devono indirizzarsi. Sei dei sette schemi di regola definiscono regole complete e uno degli schemi definisce una regola stateless.

Le regole complete correlano più eventi che si verificano durante un periodo di tempo specifico e generano una risposta a quegli eventi. Le regole non ordinate elaborano solo un evento singolo che risponde a una certa condizione e che genera una risposta a quell'evento.

regola stateful

Una regola che contiene le informazioni di stato, che sono informazioni sulle caratteristiche di un'istanza della regola, allo scopo di agire su una raccolta di eventi in un periodo di tempo. Le regole che sono definite da uno dei seguenti schemi di regola: raccolta, calcolo, duplicato, filtro, sequenza, soglia o timer.

regola stateless

Una regola che non contiene informazioni di stato e quindi può agire solo su un evento per volta. Una regola che è definita dallo schema di filtro è una regola stateless.

Riferimenti correlati

Capitolo 4, "Riepilogo sull'organizzazione della serie di regole", a pagina 47
Questo riepilogo elenca tutti gli elementi di linguaggio di una serie di regole, un blocco di regole ed ogni tipo di regola. Esso serve come riferimento rapido per la codifica di una serie di regole.

Capitolo 5, "Panoramica sugli elementi di linguaggio", a pagina 59
Questa panoramica descrive i dettagli degli elementi di linguaggio nello schema XML per il linguaggio delle regole dell'Active Correlation Technology. Gli elementi di linguaggio vengono elencati in ordine alfabetico e gli attributi che sono disponibili per ciascun elemento sono descritti all'interno della sezione per quell'elemento.

Capitolo 6, "Glossario", a pagina 121
Questo glossario contiene i termini e le definizioni di concetti importanti dell'Active Correlation Technology.

Anatomia di una regola

Le parti principali di una regola sono la selezione evento, la chiave di raggruppamento, la finestra di tempo per le regole di stato, la risposta della regola, l'intervallo di attivazione e le azioni di durata. Una regola include anche espressioni e variabili. Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola.

Selezione evento

I criteri di selezione evento determinano gli eventi che sono accettati per l'elaborazione della regola. L'elemento `<eventSelector>` definisce i criteri di selezione evento per una regola. La selezione evento si applica a tutte le regole tranne quelle che sono definite dallo schema timer. Poiché la regola timer non elabora eventi, non contiene criteri di selezione evento.

Chiave di raggruppamento

In genere ogni regola attiva ha un'istanza della regola o una copia che è in esecuzione nell'engine dell'Active Correlation Technology. Tuttavia, talvolta la stessa regola è necessaria per gruppi diversi di eventi, spesso correlati ai diversi gruppi di risorse. La chiave di raggruppamento è un metodo per indirizzare una regola in modo da creare un'istanza della regola separata (o una sua copia) per ciascun gruppo di eventi che condividono caratteristiche comuni.

La chiave di raggruppamento serve come forma ulteriore di selezione evento. Se una regola viene definita con una chiave di raggruppamento e la regola riceve un evento con le caratteristiche definite dalla chiave di raggruppamento, l'evento viene inviato all'istanza della regola che sta elaborando gli eventi che condividono quella caratteristica. Ad esempio è possibile definire una regola che raccoglie tutti gli eventi di sicurezza del tipo Audit Failure e definire la chiave di raggruppamento in modo da essere attribuito hostname di un evento. La regola può ora essere utilizzata più volte, con una copia separata della regola in esecuzione per ciascun valore univoco dell'attributo hostname. E' possibile anche controllare tutti i sistemi che ricevono l'evento Audit Failure per determinare se si verificano più di 10 eventi come questi in un periodo di tempo di 2 minuti per ogni nome host.

L'elemento `<groupingKey>` definisce la chiave di raggruppamento per una regola ed è valido per le regole che vengono definite dagli schemi di raccolta, calcolo, duplicato, sequenza e soglia.

Finestra di tempo per le regole di stato

Siccome le regole stateful correlano più eventi che si verificano durante un periodo di tempo specifico, una parte basilare di una regola stateful è la finestra di tempo definita dall'elemento `<timeWindow>`. La finestra di tempo specifica il periodo di tempo durante il quale la regola stateful è in elaborazione per abbinarsi al suo schema.

Risposta della regola

Le azioni di risposta della regola definiscono le azioni da intraprendere quando la regola completa la sua elaborazione. Ognuno dei seguenti elementi di linguaggio definisce un tipo diverso di azione di risposta della regola:

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

I tipi di azione di risposta della regola che sono disponibili per una regola dipendono dallo schema della regola.

Intervallo di attivazione

L'intervallo di attivazione definisce quando una regola è attiva e inattiva. L'elemento `<activationInterval>` definisce l'intervallo di attivazione per una regola.

Una regola può essere attivata o disattivata in una fase definita o da un evento specifico.

Se si specifica che una regola deve essere attivata o disattivata, in una fase definita e da un evento specifico, la regola viene attivata o disattivata in base a ciò che si verifica per primo, la fase o la ricezione dell'evento. Tuttavia, in questo caso, la regola potrebbe essere attivata o disattivata da molti eventi per tutta la durata. Ad esempio una regola potrebbe essere attivata da un evento, disattivata, attivata in una fase definita, disattivata di nuovo e attivata da un altro evento.

L'elemento `<activationByGroupingKey>` è un elemento contenuto nell'elemento `<activationInterval>`. L'elemento `<activationByGroupingKey>` contiene elementi che specificano gli eventi che possono attivare e disattivare un'istanza della regola definita dall'elemento `<groupingKey>`.

Azioni attinenti al ciclo di vita

Le azioni attinenti al ciclo di vita definiscono le azioni da intraprendere nelle quattro fasi principali nel ciclo di vita di una regola: caricamento, attivazione, disattivazione e scaricamento.

L'elemento `<lifeCycleActions>` contiene i seguenti elementi che definiscono queste azioni:

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

Ciclo di vita di una regola

Ogni fase del ciclo di vita di una regola può avere più cause ed effetti. Scrivendo e includendo le espressioni nelle azioni attinenti al ciclo di vita (come definito dall'elemento `<lifeCycleActions>`), un programma di scrittura di regole può definire le azioni da intraprendere in ogni fase.

Fasi del ciclo di vita di una regola

Di seguito vengono riportate le quattro fasi principali del ciclo di vita di una regola:

Caricamento

Il caricamento di una regola nell'engine dell'Active Correlation Technology in esecuzione, che innesca le azioni all'interno dell'elemento `<onLoad>`.

Attivazione

L'attivazione di una regola, che innesca le azioni all'interno dell'elemento `<onActivation>`.

Disattivazione

La disattivazione di una regola, che innesca le azioni all'interno dell'elemento `<onDeactivation>`.

Scaricamento

Lo scaricamento di una regola dall'engine Active Correlation Technology in esecuzione, che innesca le azioni all'interno dell'elemento <onUnload>.

Le fasi di attivazione e disattivazione possono verificarsi più volte nel ciclo di vita di una regola, ma le fasi di caricamento e scaricamento si verificano una sola volta.

In genere non è necessario definire le azioni attinenti al ciclo di vita. Di seguito vengono riportati esempi di definizione di un'azione attinente al ciclo di vita:

- Quando viene caricata una certa regola, è possibile creare una connessione ad un sistema esterno (come il gestore database) a cui accedere all'interno della regola. Quando questa stessa regola viene scaricata, la connessione viene chiusa e vengono eseguiti tutti i necessari processi di pulizia.
- Quando viene attivata una certa regola, è possibile verificare che certe risorse siano disponibili in quella regola.
- Quando una regola di soglia viene disattivata ma la soglia non è stata raggiunta e il periodo di tempo non è ancora finito, è possibile inoltrare un messaggio a qualcuno con queste informazioni.

Poiché l'attivazione e la disattivazione di una regola può verificarsi molte volte nel ciclo di vita, tutte le azioni codificate per queste fasi possono essere eseguite di frequente.

Cause ed effetti di ogni fase del ciclo di vita

La Tabella 1 riporta le cause e gli effetti di ogni fase del ciclo di vita.

Tabella 1. Cause ed effetti di ogni fase del ciclo di vita

Fase di durata	Cause	Effetti
Caricamento	Ognuna delle seguenti circostanze: <ul style="list-style-type: none">• Una regola o un blocco di regole viene aggiunto o sostituito, il che provoca il caricamento di altre regole.• La serie di regole viene sostituita nell'engine dell'Active Correlation Technology, che causa il caricamento delle regole nella nuova serie di regole.	Vengono eseguite le azioni nell'elemento <onLoad>.
Attivazione	La regola viene attivata. Una regola può essere attivata in uno dei seguenti modi: <ul style="list-style-type: none">• In base alle definizioni dell'elemento <activationInterval>• Tramite il metodo activate() disponibile con la variabile act_lib• Tramite le chiamate di applicazione al metodo activate() nell'engine dell'Active Correlation Technology	Se la regola non è attiva, vengono eseguite le azioni nell'elemento <onActivation>.

Tabella 1. Cause ed effetti di ogni fase del ciclo di vita (Continua)

Fase di durata	Cause	Effetti
Disattivazione	<p>La regola viene disattivata. Una regola può essere disattivata in uno dei seguenti modi:</p> <ul style="list-style-type: none"> • In base alle definizioni dell'elemento <code><activationInterval></code> tranne che per l'elemento <code><deactivateOnEvent></code> nell'elemento <code><activationByGroupingKey></code> che non causa la disattivazione • Tramite il metodo <code>deactivate()</code> disponibile con la variabile <code>act_lib</code> • Tramite le chiamate di applicazione al metodo <code>deactivate()</code> nell'engine dell'Active Correlation Technology 	<p>Se la regola è attiva, vengono eseguite le azioni nell'elemento <code><onDeactivation></code>.</p>
Scaricamento	<p>Ognuna delle seguenti circostanze:</p> <ul style="list-style-type: none"> • l'engine dell'Active Correlation Technology viene chiuso il che provoca lo scaricamento delle regole. • Una regola o un blocco di regole viene rimosso o sostituito, il che provoca lo scaricamento di vecchie regole. • La serie di regole viene rimossa o sostituita nell'engine dell'Active Correlation Technology, che causa lo scaricamento delle regole dalla vecchia serie di regole. 	<p>Se la regola è attiva, vengono eseguite le azioni nell'elemento <code><onDeactivation></code>, seguite dalle azioni nell'elemento <code><onUnload></code>. In caso contrario vengono eseguite solo le azioni nell'elemento <code><onUnload></code>.</p>

Organizzazione delle regole

Il linguaggio di regole di Active Correlation Technology organizza le regole in blocchi di regole che fanno parte di una serie di regole.

Serie di regole

La serie di regole contiene le regole, organizzate in blocchi di regole, da eseguire tramite l'engine dell'Active Correlation Technology. E' l'unità di esecuzione della regola. Ogni engine dell'Active Correlation Technology agisce solo su una serie di regole per volta.

Le regole contenute in una serie di regole vengono innescate dagli eventi inviati all'engine dell'Active Correlation Technology. Gli eventi vengono inoltrati in sequenza alle regole appropriate in base ai criteri di selezione evento e viene eseguita una regola per volta. Lo stesso evento può essere applicato e, quindi, attivato su più regole. Queste regole non sono necessariamente collegate, ma possono esserlo.

L'ordine dei blocchi di regole e delle regole all'interno della serie di regole determina come gli eventi fluttuano tramite la serie di regole.

Le variabili e le importazioni possono essere definite al livello di serie di regole per essere utilizzate nelle espressioni (il codice che contiene la logica personalizzata) per tutto l'ambito della serie di regole. L'importazione è un modo specifico del linguaggio di programmazione per accedere al codice esterno. Un programma di scrittura di regola può definire l'importazione di moduli esterni (come classi Java)

da utilizzare nelle espressioni all'interno delle regole.

Blocco di regole

Il blocco di regole è l'unità organizzativa per raggruppare le regole in base alla funzione nei domini all'interno della serie di regole. Un dominio è la categoria a cui si applica un gruppo di regole in base alla loro funzione. Ad esempio, un dominio può rappresentare un'area geografica specifica, la disciplina di gestione IT (come il rilevamento della sicurezza o la correlazione di eventi di rete) o l'organizzazione aziendale (come un'azienda o una sua divisione).

I blocchi di regole possono contenere le regole e altri blocchi di regole. Poiché i blocchi di regole possono essere concatenati, è possibile costruire una gerarchia di regole. Ad esempio una serie di regole potrebbe contenere un blocco di regole per la correlazione degli eventi di rete e il blocco di regole per la correlazione degli eventi di rete potrebbe contenere altri due blocchi di regole: uno per la correlazione di livello 2 e uno per la correlazione IP.

Di conseguenza, una serie di regole fornisce le funzioni di correlazione eventi per una varietà di domini e un blocco di regole fornisce l'organizzazione per questi diversi domini che potrebbero necessitare dell'accesso ad una serie simile di eventi.

Le variabili e le importazioni possono essere definite al livello di blocco di regole per essere utilizzate nelle espressioni per tutto l'ambito del blocco di regole. L'ambito di un blocco di regole include le regole e altri blocchi di regole contenuti in un blocco di regole.

Regola

La regola è un'unità di correlazione che deve essere utilizzata per riconoscere le relazioni tra gli eventi e per eseguire le risposte di regola appropriate. Una regola è un'implementazione di uno dei seguenti sette degli schemi di regola ed è organizzata, in base alla propria funzione, in un blocco di regole che è parte di una serie di regole:

- schema di raccolta
- schema di calcolo
- schema di duplicato
- schema di filtro
- schema di sequenza
- schema di soglia
- schema timer

Ogni regola può fornire le funzioni di correlazione eventi in base al proprio schema e le regole possono essere concatenate tramite l'inoltro eventi. Tramite questa concatenazione di regole, le funzioni di correlazione eventi di schemi diversi possono essere combinate o concatenate.

Le variabili possono essere definite al livello di regola per essere utilizzate nelle espressioni per tutto l'ambito della regola.

Riepilogo

In definitiva, la serie di regole è l'unità di esecuzione, il blocco di regole è l'unità organizzativa e la regola è l'unità di correlazione. Una serie di regole contiene uno

o più blocchi di regole, ognuno dei quali può contenere ulteriori blocchi di regole. Ognuno dei blocchi di regole contiene le regole per un dominio specifico. I blocchi di regole possono essere concatenati per costruire una gerarchia di regole. L'ordine dei blocchi di regole e delle regole all'interno della serie di regole determina come gli eventi fluttuano tramite la serie di regole.

Le variabili e le importazioni possono essere definite al livello di serie di regole per essere utilizzate nelle espressioni all'interno delle regole. L'ambito della variabile o dell'importazione è la rispettiva serie o blocco di regole. Le variabili possono anche essere definite al livello di regola, ma ciò limita l'ambito a una sola regola.

Schemi di regola

Uno schema di regola è una rappresentazione di una condizione di correlazione eventi (come una condizione di soglia o un rilevamento di evento duplicato). Il linguaggio delle regole dell'Active Correlation Technology definisce i seguenti schemi di regola: raccolta, calcolo, duplicato, filtro, sequenza, soglia e timer.

Lo schema di una regola ha valore quando si verifica la condizione definita dalla regola. Quando lo schema corrisponde, la regola conclude l'elaborazione eseguendo le azioni di risposta appropriate. Mentre è attiva una regola, lo schema di regola può essere applicato più volte.

Le regole che sono definite da uno schema di filtro sono le sole regole senza stato nel linguaggio della regola. Tutte le altre regole sono complete.

Schema di raccolta

La regola di raccolta viene definita dallo schema di raccolta. Essa raccoglie un gruppo di eventi selezionati in un intervallo di tempo. Essa è una regola stateful.

Panoramica

Lo schema di raccolta viene utilizzato per raccogliere eventi simili in un certo periodo di tempo. Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>` nella lingua della regola.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di raccolta, la risposta di regola viene eseguita quando la finestra è completa, come definita dall'elemento `<onTimeWindowComplete>`.

Esempio d'uso di questo schema di regola

Un esempio d'uso di uno schema di raccolta è una regola che funziona in questo modo:

Essa raccoglie gli eventi che rispondono ai criteri di una selezione di evento durante l'intervallo di tempo. Quando l'intervallo termina, gli eventi raccolti vengono sintetizzati in un evento singolo che contiene il conteggio totale degli eventi e delle informazioni sugli eventi sintetizzati.

Riferimenti correlati

"Riepilogo di regola Collection" a pagina 48

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di raccolta.

Schema di calcolo

Una regola di calcolo viene definita da uno schema di calcolo. Essa applica un calcolo (tramite un'espressione) agli eventi raccolti in quanto ogni evento viene ricevuto in un intervallo di tempo. Essa è una regola stateful.

Panoramica

Lo schema di calcolo esegue una funzione di calcolo, come definita dall'elemento `<computeFunction>` nel linguaggio con cui è scritta la regola, rispetto a ciascun evento accettato in un determinato periodo di tempo. Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>`.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di calcolo, la risposta di regola viene eseguita quando la finestra è completa, come definita dall'elemento `<onTimeWindowComplete>`. Il valore del calcolo è disponibile durante l'azione `<onTimeWindowComplete>`.

Esempio d'uso di questo schema di regola

Prevedere che un'applicazione stia elaborando gli eventi di ordine cliente. Un esempio d'uso di uno schema di calcolo è una regola che funziona in questo modo:

Ogni volta che viene ricevuto un evento, il valore totale dell'ordine viene aggiunto al valore totale di tutti gli ordini che si sono verificati durante il periodo di tempo specificato e il valore totale di tutti gli ordini viene pubblicato in un'interfaccia utente.

Riferimenti correlati

"Riepilogo della regola computation" a pagina 50

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di calcolo.

Schema di duplicato

Una regola di duplicato viene definita da uno schema di duplicato. Esso conta gli eventi secondi e successivi accettati all'interno di un intervallo di tempo specificato ma ignora l'elaborazione della serie di regole per questi eventi. Essa è una regola stateful.

Panoramica

Lo schema di duplicato è in genere utilizzato per isolare eventi simili (duplicati) in un periodo di tempo. Un evento duplicato è simile in qualche modo ad un evento precedente, ma non è necessariamente una copia esatta di quell'evento. Gli eventi sono duplicati semplicemente se essi rispondono ai criteri di selezione evento per la regola. Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>` nella lingua della regola.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di duplicato, la risposta della regola viene eseguita nei seguenti momenti:

- Quando viene rilevato il primo evento, come definito dall'elemento `<onDetection>`.
- Quando viene elaborato ogni evento duplicato, come definito dall'elemento `<onNextEvent>`.

- Quando la finestra di tempo è completa, come definito dall'elemento `<onTimeWindowComplete>`.

Il primo evento attiva l'azione `<onDetection>` anche se non sono stati ricevuti eventi duplicati. Il motivo di questo comportamento è che è possibile inoltrare il primo evento e ignorare l'elaborazione della serie di regole per gli eventi duplicati. In questo caso, è possibile aggiungere un'azione di risposta della regola che inoltra il primo evento quando viene attivata l'azione `<onDetection>` per la regola.

L'elaborazione predefinita per gli eventi duplicati (gli eventi secondi e susseguenti) è di conteggiare un evento duplicato ma di ignorare l'elaborazione della serie di regole per un evento duplicato. Per eseguire ulteriori azioni su un evento duplicato, è possibile definire in modo esplicito un'azione `<onNextEvent>`. Ad esempio, in certi casi, l'evento duplicato rappresenta un evento che può già essere registrato in un database o in altro contenitore. Di conseguenza, è possibile codificare un'azione `<onNextEvent>` per rimuovere l'evento duplicato da queste altre ubicazioni.

Un'azione `<onTimeWindowComplete>` può essere utilizzata per creare un record di riepilogo per tutti gli eventi duplicati che includa il numero di duplicati elaborati.

Esempio d'uso di questo schema di regola

Considerare che il messaggio "Servizio negato" continui ad essere emesso dallo stesso tipo di risorsa (un monitor della sicurezza). Ciò indica una possibile violazione della sicurezza. Un esempio d'uso di uno schema di duplicato è una regola che funziona in questo modo:

Dopo l'emissione di un messaggio "Servizio negato" dal monitor della sicurezza, tutti i duplicati di quell'evento che si verificano durante un periodo di tempo di 30 secondi vengono conteggiati ma non inviati alla console dell'operatore. Inoltre, alla fine del periodo di tempo di 30 secondi, la regola genera un evento che indica il numero di messaggi "Servizio negato" che si sono verificati in quel periodo di tempo.

Riferimenti correlati

"Riepilogo regola di duplicato" a pagina 51

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di duplicato.

Schema di filtro

La regola di filtro viene definita dallo schema di filtro. Essa intraprende una certa azione quando accetta un evento. Essa agisce su un singolo evento ed è perciò una regola senza definizione di stato.

Panoramica

Lo schema di filtro viene utilizzato per agire su eventi singoli che rispondono ai criteri di selezione evento. Diversamente dagli altri schemi di regola, non vengono mantenute le informazioni di stato associate (come la cronologia degli eventi precedenti).

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di filtro, la risposta di regola viene eseguita quando vengono ricevuti tutti gli eventi che rispondono ai criteri di selezione evento, come definito dall'elemento `<onDetection>`.

Esempio d'uso di questo schema di regola

Un esempio d'uso di uno schema di filtro è una regola che funziona in questo modo:

Se un evento `ServerStatus` indica un `serverLoad` maggiore del 95%, la regola esegue un'azione di segnalazione all'amministratore.

Riferimenti correlati

"Riepilogo di regola di filtro" a pagina 52

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di filtro.

Schema di sequenza

Una regola di sequenza viene definita da uno schema di sequenza. Essa rileva se una certa sequenza di eventi arriva all'interno di un intervallo di tempo. La sequenza può essere ordinata o casuale. Essa è una regola stateful.

Panoramica

Lo schema di sequenza ricerca una sequenza di eventi in un certo periodo di tempo e rileva se la sequenza è completa o incompleta. Una sequenza incompleta include uno o più eventi nell'ordine specificato ma non tutti.

Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>` nella lingua della regola. Ogni evento nella sequenza è definito da un elemento `<eventSelector>` separato all'interno della regola. La sequenza di eventi può essere rilevata in uno dei seguenti ordini:

- Nell'ordine in cui gli elementi `<eventSelector>` sono codificati per la regola. In questo caso, quando la regola rileva l'evento che è definito dal primo elemento `<eventSelector>`, viene avviato il rilevamento della sequenza. La regola attende poi l'evento che viene definito dal secondo elemento `<eventSelector>`.
- In ordine casuale. In questo caso, quando la regola rileva l'evento che è definito dagli elementi `<eventSelector>`, viene avviato il rilevamento della sequenza. La regola attende poi un altro degli eventi che sono definiti dagli elementi `<eventSelector>`.

Lo schema di sequenza è differente da altri schemi di regola nei seguenti modi principali:

- Esso ha più elementi `<eventSelector>` per definire quali eventi sono accettati dalla regola. Esso richiede anche un minimo di due elementi `<eventSelector>`.
- Quando un evento risponde ai criteri definiti da uno degli elementi `<eventSelector>`, quell'elemento `<eventSelector>` viene escluso da una ulteriore elaborazione di evento all'interno di quell'istanza della regola.
- L'attributo `alias` sull'elemento `<eventSelector>` è valido solo in una regola di sequenza e definisce in modo univoco un evento selezionato in base ad un certo programma di selezione eventi in una regola di sequenza. In un'espressione all'interno di un predicato o di un'azione di filtro, è possibile utilizzare la variabile `act_eventList` con questo nome per accedere a un evento.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di sequenza, la risposta di regola viene eseguita nei seguenti momenti:

- Quando viene rilevata la sequenza completa di eventi nella finestra di tempo, come definito dall'elemento `<onDetection>`.
- Quando vengono accettati uno o più eventi ma la sequenza completa non arriva nella finestra di tempo, come definito dall'elemento `<onTimeOut>`.

Lo schema di sequenza può risultare utile per rilevare che una certa sequenza è incompleta. Ad esempio, se si verifica un evento di "sistema guasto" senza un evento di "sistema funzionante" susseguente, il programma di scrittura di regola può codificare un'azione `<onTimeOut>` per gestire questo tipo di evento mancante.

Esempio d'uso di questo schema di regola

Scenario che illustra il rilevamento di sequenza completa:

In un ambiente IT, un amministratore desidera sapere se il valore della dimensione del pacchetto DB2 sta condizionando WebSphere Application Server e, in questo caso, desidera correggere il problema. Di conseguenza, se gli eventi successivi si verificano nel seguente ordine durante un periodo di tempo specificato, l'amministratore desidera incrementare il valore della dimensione del pacchetto DB2 e riavviare il gestore del database:

1. Eccezione di assegnazione risorsa di WebSphere Application Server. Evento di tipo `WASResourceAllocationException`.
2. Messaggio di errore DB2 che indica: "Pacchetto insufficiente per elaborare l'istruzione". Evento del tipo `DB2NotEnoughHeap`.

Due elementi `<eventSelector>` sono definiti in una regola di sequenza e gli eventi devono arrivare nell'ordine in cui gli elementi `<eventSelector>` vengono codificati per la regola (piuttosto che in ordine casuale). Il primo elemento `<eventSelector>` ricerca l'evento `WASResourceAllocationException` e il secondo elemento `<eventSelector>` ricerca l'evento `DB2NotEnoughHeap`. I seguenti eventi vengono presentati al sistema nella finestra di tempo specificata:

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

Il comportamento della regola è il seguente:

1. Il primo evento, `WASResourceAllocationException`, viene accettato. Poiché i criteri per il primo elemento `<eventSelector>` sono soddisfatti, il primo elemento `<eventSelector>` viene escluso da una ulteriore elaborazione all'interno della regola.
2. Il secondo evento, `DB2BackupStarted`, viene ignorato.
3. Il terzo evento, `WASResourceAllocationException`, viene ignorato.
4. Il quarto evento, `WASResourceAllocationException`, viene ignorato.
5. Il quinto evento, `DB2NotEnoughHeap`, viene accettato e completa la sequenza. Viene eseguita l'azione di risposta alla regola `<onDetection>`. Questa azione viene definita per incrementare il valore della dimensione del pacchetto DB2 e riavviare il gestore database. La regola ritorna allo stato iniziale.

Il primo elemento `<eventSelector>` viene poi incluso nell'elaborazione eventi successiva con questa regola.

Scenario che illustra il rilevamento di sequenza incompleta:

L'azienda intende avere gli ordini di tutti i clienti pronti per la spedizione in un'ora dalla ricezione dell'ordine e desidera conoscere quando ciò non si verifica.

Due elementi `<eventSelector>` sono definiti in una regola di sequenza e gli eventi devono arrivare nell'ordine in cui gli elementi `<eventSelector>` vengono codificati (piuttosto che in ordine casuale). Il primo elemento `<eventSelector>` ricerca l'evento Netsales con `operationType=Order` e il secondo elemento `<eventSelector>` ricerca l'evento Netsales con `operationType=Delivery`. I seguenti eventi vengono presentati al sistema nella finestra di tempo specificata di 1 ora:

1. Un evento Netsales con `operationType=Order`
2. Un evento Netsales con `operationType=Order`

Il comportamento della regola è il seguente:

1. Il primo evento viene accettato. Poiché i criteri per il primo elemento `<eventSelector>` sono soddisfatti, il primo elemento `<eventSelector>` viene escluso da una ulteriore elaborazione all'interno della regola.
2. Il secondo evento viene ignorato.
3. Poiché un evento Netsales con `operationType=Delivery` non viene ricevuto nell'intervallo di tempo specificato, viene eseguita la risposta di regola `<onTimeOut>`. Questa azione viene definita per notificare a un responsabile aziendale che l'ordine del cliente non è pronto per la spedizione in 1 ora dalla richiesta dell'ordine. La regola ritorna allo stato iniziale.

Il primo elemento `<eventSelector>` viene poi incluso nell'elaborazione eventi successiva con questa regola.

Concetti correlati

"Accesso alle informazioni correlate all'evento" a pagina 23

Gli esempi che seguono indicano come un programma di scrittura di regole può accedere alle informazioni correlate agli eventi tramite le variabili fornite da Active Correlation Technology.

Riferimenti correlati

"Riepilogo della regola di sequenza" a pagina 53

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di sequenza.

Schema di soglia

La regola di soglia viene definita dallo schema di soglia. Essa raccoglie un gruppo di eventi selezionati in un intervallo di tempo e determina, dopo che è stato ricevuto ogni evento, se viene soddisfatta una condizione di soglia. Essa è una regola stateful.

Panoramica

Lo schema di soglia raccoglie gli eventi in un certo periodo di tempo fino a che non viene soddisfatto un valore di soglia. Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>` nella lingua della regola.

Lo schema di soglia fornisce le seguenti tre opzioni per un tipo di soglia:

soglia di conteggio eventi

Con questo tipo di soglia, è possibile definire il numero di eventi che devono rispondere ai criteri di selezione eventi in un certo periodo di

tempo. Il valore di soglia definito viene confrontato con il numero degli eventi accettati. Quando il conteggio degli eventi è uguale al limite definito all'interno della finestra di tempo, la soglia viene soddisfatta.

Questo tipo di soglia può essere utile per un controllo molto semplice del conteggio eventi. Ad esempio, esso risponde alla domanda: "Impostare 5 eventi di errore di collegamento in un minuto?"

Questa soglia viene definita dall'elemento <eventCountThreshold>. L'elemento <eventCountThreshold> specifica anche uno dei seguenti due possibili modi di intervallo di tempo per la finestra di tempo:

intervallo fisso

Un intervallo fisso inizia quando viene ricevuto il primo evento che risponde ai criteri di selezione evento e termina quando si verifica una delle condizioni di seguito riportate.

- La regola raggiunge la soglia nella durata di tempo specificata.
- La durata di tempo specificata è trascorsa.

intervallo di scorrimento

Un intervallo scorrevole inizia quando viene ricevuto il primo evento che risponde ai criteri di selezione evento. Tuttavia quando la regola non raggiunge la soglia e la durata specificata è trascorsa, la finestra di tempo regola l'ora di inizio sull'ora di ricezione evento per un nuovo "primo" evento che è in genere l'evento successivo che viene accettato. L'intervallo scorrevole continua a regolarsi in questo modo fino a che non si verifica una delle seguenti condizioni:

- La regola raggiunge la soglia nella durata di tempo specificata.
- Dopo che viene ricevuto l'evento che avvia la finestra di tempo, non vengono ricevuti eventi susseguenti all'interno della durata di tempo specificata.

L'evento che apre la finestra di tempo (diventa il nuovo "primo" evento) è l'evento con un'ora di ricezione che risponde a questi criteri: l'ora di ricezione, aggiunta alla durata dell'intervallo di tempo, è successiva all'ora corrente. Di seguito sono riportati i criteri nel formato di un'equazione:

*ora di ricezione evento + durata di intervallo
di tempo per la regola > ora corrente*

Quando non è presente alcun evento di questo tipo, l'intervallo scorrevole non è in grado di regolare l'ora in alcun modo e l'intervallo termina.

soglia calcolata

Con questo tipo di soglia, è possibile scrivere il codice (o utilizzare il codice scritto da qualcun altro) che esegue un calcolo su ogni evento accettato e restituisce un valore di soglia calcolata che viene mantenuto in una variabile definita in precedenza. Questo valore di soglia calcolato viene poi confrontato con un valore di soglia definito per determinare se la soglia viene soddisfatta.

Di conseguenza, è possibile applicare un calcolo complesso per creare (o aggiornare) un valore di soglia calcolato, utilizzando possibilmente i dati salvati da eventi precedenti. Inoltre, il programma di scrittura eventi può impostare il valore di soglia definito indipendentemente dalla logica che calcola il valore di soglia elaborato.

Questo tipo di soglia può essere utile per l'aggregazione e il confronto di un valore con un valore di soglia definito. Ad esempio, può essere utilizzato per calcolare la somma dell'importo delle vendite ad un certo cliente in un dato periodo di tempo e per confrontare questa somma con un valore di soglia definito.

Questa soglia viene definita dall'elemento `<computedThreshold>`.

soglia booleana

Con questo tipo di soglia, è possibile scrivere il codice (o utilizzare il codice scritto da qualcun altro) che restituisce un valore `true` o `false` per ogni evento accettato. Se il valore è `true` la soglia è stata rispettata. Se il valore è `false`, la regola di soglia continua l'elaborazione fino a che il periodo non termina o finché non viene accettato un altro evento.

Questo tipo di soglia può essere utile per un controllo di una serie di valori. Ad esempio se l'uso CPU deve essere tra 30% e 80% sempre, questa soglia può verificare costantemente che l'uso rimanga all'interno di quell'intervallo.

Questa soglia viene definita dall'elemento `<booleanThreshold>`.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema di soglia, la risposta di regola viene eseguita nei seguenti momenti:

- Quando la soglia viene rispettata, come definito dall'elemento `<onDetection>`.
- Quando vengono accettati uno o più eventi ma la soglia non viene rispettata nella finestra di tempo, come definito dall'elemento `<onTimeOut>`.

Esempio d'uso di questo schema di regola

Un esempio d'uso di uno schema di soglia è una regola che funziona in questo modo:

Se vengono originati 4 eventi di server non raggiungibile dalla stessa sottorete all'interno di un intervallo di tempo di scorrimento di 30 secondi, la regola esegue un'azione per controllare lo stato di un router.

Riferimenti correlati

"Riepilogo della regola di soglia (Threshold)" a pagina 55

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di soglia.

Schema timer

La regola timer viene definita dallo schema timer. Essa avvia le azioni ad intervalli regolari. Essa è una regola stateful. Sebbene una regola timer non elabora gli eventi, essa può essere attivata o disattivata da un evento.

Panoramica

Lo schema timer è analogo a un timer che parte all'inizio di un certo periodo di tempo e termina alla fine del periodo stesso. Il periodo è indicato da una finestra definita dall'elemento `<timeWindow>` nella lingua della regola.

Se non viene specificato di non ripetere, lo schema timer viene ripetuto fino a che non viene disattivata la regola timer. Di conseguenza quando viene avviata la regola timer, essa attende il periodo di tempo specificato prima di avviare qualsiasi

azione e ripete questo comportamento fino a che non viene disattivato o fino a che non viene chiuso l'engine dell'Active Correlation Technology.

La regola timer è univoca in quanto non contiene i criteri di selezione evento. La regola timer avvia l'elaborazione in base all'intervallo di attivazione per la regola, come definito dall'elemento <activationInterval>. Se l'elemento <activationInterval> predefinito viene utilizzato e lo schema timer viene impostato per la ripetizione, la regola timer viene avviata quando viene caricata dall'engine Active Correlation Technology e si arresta quando viene chiuso l'engine dell'Active Correlation Technology. Per attivare una regola timer con un evento, è necessario specificare l'evento nell'elemento <activateOnEvent> all'interno dell'elemento <activationInterval> per la regola.

Condizioni in cui viene eseguita la risposta della regola

Con lo schema timer, la risposta di regola viene eseguita quando la finestra è completa, come definita dall'elemento <onTimeWindowComplete>.

Esempio d'uso di questo schema di regola

Lo schema timer può risultare utile per l'implementazione delle regole di pulizia. Un esempio d'uso di uno schema timer è una regola che funziona in questo modo:

Ogni 30 minuti, la regola esegue un'azione che azzerà gli eventi non rilevanti e informativi aperti per più di 48 ore.

Riferimenti correlati

"Riepilogo di regola Timer" a pagina 56

Questo riepilogo elenca tutti gli elementi di linguaggio della regola timer.

Aspetti comuni e univoci di schemi di regola differenti

Questa matrice fornisce una panoramica ad alto livello degli aspetti comuni e univoci dei diversi schemi di regola.

Tabella 2 elenca gli elementi di linguaggio primari per le regole e riporta una X nella colonna per ciascun tipo di regola in cui è valido l'elemento. Gli elementi di linguaggio primari sono elementi secondari diretti dei diversi tipi di regola. Questo elenco non include gli elementi che sono contenuti all'interno degli elementi secondari diretti, che possono anche variare secondo il tipo di regola. Inoltre, la validità di alcuni attributi di elemento può variare secondo il tipo di regola.

Tabella 2. Matrice di aspetti comuni e univoci dei diversi schemi di regola

Elemento	raccolta	calcolo	duplicato	filtro	sequenza	soglia	timer
<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	

Tabella 2. Matrice di aspetti comuni e univoci dei diversi schemi di regola (Continua)

Elemento	raccolta	calcolo	duplicato	filtro	sequenza	soglia	timer
<onNextEvent>			X				
<onTimeOut>					X	X	
<onTimeWindowComplete>	X	X	X				X

Espressioni

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Gli scrittori della regola possono codificare le espressioni per scopi diversi secondo il contesto e il risultato che essi desiderano ottenere. Le espressioni vengono utilizzate di frequente per l'inizializzazione delle variabili, per la definizione dei criteri di selezione eventi e per la specifica della risposta della regola e delle azioni attinenti al ciclo di vita.

Elementi di lingua che contengono le espressioni

Ogni elemento di lingua che contiene un'espressione ha un attributo `expressionLanguage` che identifica il linguaggio di programmazione in cui è scritta l'espressione. Il linguaggio di programmazione Java è l'unico linguaggio di espressione supportato.

Le espressioni possono essere contenute nei seguenti elementi del linguaggio della regola.

- <varInitializer> per una serie di regole, blocco di regole o variabile di regola
- <filteringPredicate> su <eventSelector>
- <computedValue> su <groupingKey>
- <computeFunction> su una regola di calcolo
- <booleanThreshold> su una regola di soglia
- <computedThreshold> su una regola di soglia
- Le azioni di risposta della regola per una regola:
 - <action> in <onDetection>. Quest'azione è valida solo per le regole di duplicato, filtro, sequenza e soglia.
 - <action> in <onNextEvent>. Quest'azione è valida solo per la regola di duplicato.
 - <action> in <onTimeOut>. Quest'azione è valida solo per le regole di sequenza e soglia.
 - <action> in <onTimeWindowComplete>. Quest'azione è valida solo per le regole di raccolta, calcolo, duplicato e timer.
- Azioni di durata per una regola:
 - <action> in <onLoad>
 - <action> in <onActivation>
 - <action> in <onDeactivation>
 - <action> in <onUnload>

Supporto dell'Active Correlation Technology nella codifica delle espressioni

Come supporto alla codifica di espressioni, Active Correlation Technology fornisce le seguenti funzioni:

- Importazione di moduli esterni (come classi Java) da utilizzare nelle espressioni.
- Inizializzazione e accesso alla serie di regole, blocco di regole o variabili di regola.
- Tramite la variabile `act_event`, accesso all'evento corrente che una regola sta elaborando.
- Tramite la variabile `act_eventCount`, accesso al numero di eventi che sono stati accettati da una regola.
- Tramite la variabile `act_eventList`, accesso all'elenco degli eventi che sono stati accettati da una regola. Ciò include la potenzialità di accesso ai diversi attributi di un evento e la potenzialità di accesso ad ogni evento in una regola di sequenza tramite il relativo nome alias.
- Tramite la variabile `act_lib`, accesso ai metodi che includono la potenzialità di richiesta e impostazione delle variabili e di controllo del flusso di eventi tramite una serie di regole.
- Tramite la variabile `act_location` accesso all'ubicazione, all'interno della gerarchia di regole di un'espressione.
- Tramite la variabile `act_nodeName`, accesso al nome completo di un nodo.
- Tramite la variabile `act_threshold`, accesso al valore di soglia per una regola di soglia.

Importazione e accesso ai moduli e oggetti esterni

Questo esempio indica come è possibile rendere il codice esterno (come le classi Java) e gli oggetti esterni accessibili alle espressioni. Un oggetto esterno è un oggetto che un'applicazione crea per comunicare con le espressioni.

Prima di poter accedere al codice di accesso esterno da un'espressione, è necessario rendere il codice accessibile alle espressioni.

L'importazione è un modo specifico del linguaggio di programmazione per rendere il codice esterno accessibile alle espressioni. L'elemento `<import>` contiene un tipo speciale di espressione che specifica i moduli esterni (come le classi Java) da importare per essere utilizzati in altre espressioni all'interno delle regole. Un'importazione può essere definita sul livello della serie di regole o di un blocco di regole.

Il seguente elemento `<import>` contiene un'espressione, scritta nel linguaggio di programmazione Java, che importa la classe `StaticHelper` e la classe `Queue` che può essere indicata da altre espressioni:

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

Sebbene l'uso del nome classe completo non è richiesto nell'istruzione `import`, è necessario specificare il nome completo per evitare un tempo di compilazione esteso. Ad esempio la classe Java deve essere specificata come `com.ibm.act.sample.StaticHelper` piuttosto che come `com.ibm.act.sample.*` o `com.ibm.act.*`.

Accesso a un metodo statico

L'esempio che segue indica come un'espressione all'interno dell'azione di risposta della regola fa riferimento alla classe StaticHelper dopo che la classe è stata importata:

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

Accesso a un metodo di istanza per un oggetto

L'esempio che segue indica come un'espressione all'interno dell'azione di risposta della regola fa riferimento alla classe Queue dopo che la classe è stata importata. In questo esempio, si ottiene un oggetto esterno con il nome di OutputQueueOne e un tipo di Queue e si utilizza per inserire un evento in una coda specifica.

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

Inizializzazione e accesso alle variabili

Questo esempio indica come inizializzare e accedere ad una serie di regole, un blocco di regole o a delle variabili di regola.

Una variabile può essere definita al livello della serie di regole, di un blocco di regole o di una regola. Prima di poter accedere a una variabile, essa deve essere inizializzata con un'espressione di inizializzazione. La seguente espressione inizializza due variabili, hostsList e 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>
```

A tutte le variabili è possibile accedere tramite le espressioni. L'esempio che segue mostra come si accede alle variabili hostsList e hostsString, che sono state inizializzate nell'esempio precedente, tramite un'espressione all'interno di un'azione di risposta di regola. In questo esempio, hostsList viene modificato e a hostsString viene fornito un nuovo valore.

```
<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", newHostString);
  </action>
</onNextEvent>
```

Accesso alle informazioni correlate all'evento

Gli esempi che seguono indicano come un programma di scrittura di regole può accedere alle informazioni correlate agli eventi tramite le variabili fornite da Active Correlation Technology.

Esempio di accesso all'evento corrente:

Il seguente codice mostra come utilizzare la variabile `act_event` per ottenere l'attributo `hostname` di un evento:

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

Esempio di accesso agli eventi tramite l'elenco di eventi:

Il seguente codice mostra come utilizzare la variabile `act_eventList` per ottenere il primo evento dell'elenco:

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

Esempio di accesso agli eventi tramite l'elenco di eventi in base a alias:

Diversamente dagli altri tipi di regola, la regola di sequenza consente più programmi di selezione eventi e richiede, in effetti, un minimo di due programmi di selezione eventi. L'attributo `alias` sull'elemento `<eventSelector>` è valido solo in una regola di sequenza e definisce in modo univoco un evento selezionato in base ad un certo programma di selezione eventi in una regola di sequenza. In un'espressione all'interno di un predicato o di un'azione di filtro, è possibile utilizzare la variabile `act_eventList` con questo nome per accedere a un evento.

Il codice che segue mostra due programmi di selezione eventi per una regola di sequenza. I nomi `alias` sono `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>
```

Il seguente codice mostra come utilizzare la variabile `act_eventList` per ottenere l'evento che è stato accettato dal primo programma di selezione evento, chiamato `TECevent`:

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

Procedure ottimali per le espressioni di codifica

Queste informazioni includono alcune procedure ottimali, consigli e suggerimenti per codificare le espressioni in modo efficiente ed efficace

- Per facilitare la comprensione, la maggior parte degli esempi di espressione forniti in queste informazioni, includono il codice Java direttamente nei costrutti XML. Tuttavia quando vengono create le regole, il metodo migliore è di utilizzare i moduli esterni per contenere il codice Java e richiamare i moduli esterni come parti delle espressioni.

E' anche possibile utilizzare frammenti esistenti o crearne dei nuovi, nella costruzione di regole per fornire il codice per le chiamate ai moduli esterni. I frammenti sono estratti di codice sorgente utilizzabili nelle espressioni. Nella costruzione di regole, i frammenti sono disponibili tramite la vista `Frammenti`.

Con questo approccio, le attività di progettazione, sviluppo, modifica, prova e debug del codice Java possono essere eseguite in un ambiente IDE (integrated development environment) a scelta e può essere controllato come parte regolare del processo di sviluppo.

- Per evitare l'analisi del codice di espressione come XML, racchiudere il codice in una sezione CDATA, in cui `<![CDATA[` precede immediatamente il codice e `]]>` lo segue, come nel seguente esempio:

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

I programmi di analisi XML ignorano tutto ciò che è all'interno di una sezione CDATA.

- Se si utilizza il metodo `act_lib.getExternalContext()` in un'espressione, non memorizzare l'oggetto restituito dal metodo in una variabile di una serie di regole o di un blocco di regole. Il motivo è che un'applicazione può modificare il riferimento per l'istanza dell'oggetto e la variabile di serie di regole o di blocco di regole associata non viene aggiornata.
- Se si utilizza un'istruzione di restituzione (`return;`) in un'espressione all'interno dell'elemento `<action>` e ulteriori elementi `<action>` vengono codificati rispettivamente per l'azione di risposta della regola o per l'azione attinente al ciclo di vita, l'esecuzione del codice termina laddove l'istruzione `return` viene codificata e inizia di nuovo nell'espressione all'interno dell'elemento successivo `<action>`.
- La gestione delle regole e altri metodi dell'engine dell'Active Correlation Technology possono essere richiamati all'interno dell'azione di risposta della regola o all'interno dell'azione attinente al ciclo di vita.

Informazioni correlate

"Importazione e accesso ai moduli e oggetti esterni" a pagina 21

Questo esempio indica come è possibile rendere il codice esterno (come le classi Java) e gli oggetti esterni accessibili alle espressioni. Un oggetto esterno è un oggetto che un'applicazione crea per comunicare con le espressioni.

"Inclusione di frammenti nelle espressioni all'interno delle regole" a pagina 42

Nelle espressioni all'interno delle regole, è anche possibile includere i frammenti dalla vista Frammenti di Eclipse Workbench.

Variabili

Nel linguaggio della regola, certe variabili vengono utilizzate per memorizzare le informazioni correlate agli eventi in diverse ricorrenze o regole. A queste informazioni correlate all'evento è possibile accedere dalle espressioni all'interno delle regole. Alcuni tipi di variabili vengono definiti dal programma di scrittura della regola e altri sono forniti dall'Active Correlation Technology. Ad alcuni tipi è possibile accedere direttamente all'interno di un'espressione ed ad altri solo tramite i metodi che sono forniti dall'Active Correlation Technology.

Variabili definite negli elementi <variable> e da accedere tramite metodi

E' possibile definire una variabile all'interno dell'elemento <variable> per una regola, un blocco di regole o una serie di regole. E' possibile accedere a questa variabile all'interno di un'espressione utilizzando uno dei seguenti metodi:

- Il metodo `getVariable()` o uno dei metodi `getjavatypeVariable()`
- Il metodo `setVariable()` o uno dei metodi `setjavatypeVariable()`

Ad esempio, se viene definita la variabile `rule_writer_variable` all'interno dell'elemento <variable> per una regola, è possibile accedere a questa variabile con il seguente codice:

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

Variabili fornite da Active Correlation Technology e con accesso diretto in un'espressione

Le seguenti variabili vengono fornite dall'Active Correlation Technology. E' possibile utilizzare queste variabili in linea all'interno di un'espressione.

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

Ad esempio, con il seguente codice, è possibile accedere alla variabile `act_event` per ottenere l'attributo `hostname` di un evento:

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

Variabili fornite da Active Correlation Technology e con accesso tramite metodi

Le seguenti variabili vengono fornite dall'Active Correlation Technology. E' possibile accedere a queste variabili in un'espressione utilizzando il metodo `getVariable()` o uno dei metodi `getjavatypeVariable()`.

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

Ad esempio, con il seguente codice, è possibile accedere alla variabile `act_eventCount`:

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

Tabella 3 riporta le costanti che l'interfaccia `IACTLibrary` fornisce per queste variabili. Nel proprio codice, per assicurare che errori di ortografia o tipografici vengano rilevati durante la compilazione piuttosto che durante l'esecuzione, utilizzare sempre le costanti che rappresentano queste variabili piuttosto che le variabili stesse. Ad esempio utilizzare `act_lib.getIntVariable(IACTLibrary.EVENTCOUNT);` piuttosto che `act_lib.getIntVariable("act_eventCount");`.

Tabella 3. Variabili con costanti associate

Variabile	Costante associata
<code>act_eventCount</code>	<code>EVENTCOUNT</code>

Tabella 3. Variabili con costanti associate (Continua)

Variabile	Costante associata
act_location	LOCATION
act_nodeName	NODENAME
act_threshold	THRESHOLD

Riferimenti correlati

“Elemento variable” a pagina 116

L’elemento <variable> definisce una variabile e contiene le informazioni in un formato che può essere indicato dalle espressioni. Una variabile può essere definita al livello della serie di regole, di un blocco di regole o di una regola.

Tipi di dati per le variabili di Active Correlation Technology

Le variabili fornite da Active Correlation Technology hanno diversi tipi di dati.

Tabella 4 mostra i tipi di dati per queste variabili.

Tabella 4. Tipi di dati per le variabili di Active Correlation Technology

Variabile	Tipo dati
act_event	Il tipo definito dall’interfaccia com.ibm.correlation.IEvent
act_eventCount	int
act_eventList	Il tipo definito dall’interfaccia com.ibm.correlation.IEventList
act_lib	Il tipo definito dall’interfaccia com.ibm.correlation.IACTLibrary
act_location	java.lang.String
act_nodeName	java.lang.String
act_threshold	Poiché la variabile è il valore dell’attributo di soglia dell’elemento <computedThreshold> o <eventCountThreshold> di una regola di soglia, il tipo di dati deve essere lo stesso di quello relativo al valore dell’attributo di soglia.

Contesti di espressione in cui sono valide le variabili

Le variabili fornite da Active Correlation Technology sono valide sono in contesti di espressione specifici.

Tabella 5 elenca i contesti di espressione in cui queste variabili sono valide. La tabella riporta una X nella colonna per ciascuna variabile valida all’interno del relativo contesto di espressione. Le altre limitazioni d’uso che si applicano a queste variabili vengono descritte nella sezione della rispettiva variabile.

Tabella 5. Contesti di espressione in cui sono valide le variabili

Contesto espressione	act_event	act_eventCount	act_eventList	act_lib	act_location	act_nodeName	act_threshold
<action> in <onActivation>	X			X	X	X	X
<action> in <onDeactivation>	X	X	X	X	X	X	X
<action> in <onDetection>	X	X	X	X	X	X	X
<action> in <onLoad>				X	X	X	X
<action> in <onNextEvent>	X	X	X	X	X	X	
<action> in <onTimeOut>		X	X	X	X	X	X
<action> in <onTimeWindowComplete>		X	X	X	X	X	
<action> in <onUnload>				X	X	X	X

Tabella 5. Contesti di espressione in cui sono valide le variabili (Continua)

Contesto espressione	act_event	act_eventCount	act_eventList	act_lib	act_location	act_nodeName	act_threshold
<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> per una regola				X	X	X	X

Variabile act_event

La variabile act_event fornisce l'accesso ai metodi che si applicano all'evento corrente.

Dettagli

Poiché una regola di timer non elabora eventi, la variabile act_event in una regola di timer si applica solo agli eventi che attivano o disattivano la regola.

La variabile act_event si applica nelle azioni <onActivation> e <onDeactivation> solo se un evento ha attivato o disattivato la regola. In caso contrario, questa variabile è nulla.

Esempio di codifica

Il seguente codice ha accesso alla variabile act_event per ottenere l'attributo hostname di un evento:

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

Metodi a cui è possibile accedere

I metodi a cui la variabile act_event fornisce l'accesso sono definiti nell'interfaccia IEvent, come riportato in Tabella 6 a pagina 28.

Tabella 6. Interfaccia IEvent con metodi corrispondenti e ubicazione delle descrizioni di metodo Javadoc

Interfaccia	Metodi	Ubicazione delle descrizioni del metodo 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

Variabile act_eventCount

La variabile act_eventCount è un numero intero uguale al numero degli eventi che sono stati accettati da una regola.

Dettagli

Per una regola di duplicato, il valore della variabile act_eventCount è il numero totale degli eventi accettati, che include gli eventi di origine e i duplicati. Per tutti gli altri tipi di regola, il valore è lo stesso della dimensione dell'elenco di eventi, che può essere ottenuto tramite la variabile act_eventList utilizzando il metodo act_eventList.size().

Le variabili act_eventCount e act_eventList non sono valide all'interno di una regola di timer in quanto questa non elabora eventi.

Se una regola viene definita con una chiave di raggruppamento, le variabili act_eventCount, act_eventList e act_threshold non sono valide all'interno dei seguenti contesti di espressione:

- Azioni attinenti al ciclo di vita
- <filteringPredicate> in <activateOnEvent> o <deactivateOnEvent> in <activationInterval>
- <computedValue>

Ciò in quanto, in questo caso, le variabili di regola si applicano solo a un'istanza della regola e le istanze della regola non esistono al momento in cui queste espressioni vengono eseguite.

Esempio di codifica

Con il seguente codice si accede alla variabile act_lib per ottenere il numero di eventi che sono stati accettati da una regola:

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

Variabile act_eventList

La variabile act_eventList fornisce l'accesso ai metodi che si applicano all'elenco di eventi che sono stati accettati da una regola.

Dettagli

Una regola di filtro e una regola di duplicato hanno sempre un elenco di non più di un evento in quanto una regola di filtro è una regola stateless e una regola di duplicato mantiene solo il primo evento analizzato.

Le variabili act_eventCount e act_eventList non sono valide all'interno di una regola di timer in quanto questa non elabora eventi.

Se una regola viene definita con una chiave di raggruppamento, le variabili act_eventCount, act_eventList e act_threshold non sono valide all'interno dei seguenti contesti di espressione:

- Azioni attinenti al ciclo di vita
- <filteringPredicate> in <activateOnEvent> o <deactivateOnEvent> in <activationInterval>
- <computedValue>

Ciò in quanto, in questo caso, le variabili di regola si applicano solo a un'istanza della regola e le istanze della regola non esistono al momento in cui queste espressioni vengono eseguite.

Esempio di codifica

Il seguente codice ha accesso alla variabile act_eventList per ottenere il secondo evento dell'elenco:

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

Metodi a cui è possibile accedere

I metodi a cui la variabile act_eventList fornisce l'accesso sono definiti nell'interfaccia IEventList, come riportato in Tabella 7.

Tabella 7. Interfaccia IEventList con metodi corrispondenti e ubicazione delle descrizioni del metodo Javadoc

Interfaccia	Metodi	Ubicazione delle descrizioni del metodo Javadoc
IEventList	<ul style="list-style-type: none">• get• size• isEmpty• listIterator	com.ibm.correlation.IEventList

Variabile act_lib

La variabile act_lib fornisce l'accesso ai metodi di libreria in Active Correlation Technology.

Dettagli

I metodi a cui la variabile `act_lib` può accedere variano a seconda dell'elemento del linguaggio di regola che contiene l'espressione in cui viene utilizzata la variabile. Vedi Tabella 8.

Tabella 8. I metodi a cui la variabile `act_lib` può accedere si basano sul contesto dell'espressione che la contiene.

Contesto espressione	Metodi IACTLibrary	Metodi IExitableActionLibrary	Metodi IActionLibrary
<action> in <onActivation>	X		
<action> in <onDeactivation>	X		
<action> in <onDetection>	X	X	X
<action> in <onLoad>	X		
<action> in <onNextEvent>	X	X	X
<action> in <onTimeOut>	X		X
<action> in <onTimeWindowComplete>	X		X
<action> in <onUnload>	X		
<booleanThreshold>	X		
<computedThreshold>	X		
<computedValue>	X		
<computeFunction>	X		
<filteringPredicate>	X		
<varInitializer>	X		

Esempio di codifica

Il seguente codice ha accesso alla variabile `act_lib` per richiamare il metodo con cui si esce dalla serie di regole:

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

Metodi a cui è possibile accedere

I metodi a cui la variabile `act_lib` fornisce l'accesso sono definiti nelle interfacce `IACTLibrary`, `IExitableActionLibrary` e `IActionLibrary`, come riportato in Tabella 9 a pagina 31.

Tabella 9. Interfacce con metodi corrispondenti e ubicazione delle descrizioni del metodo Javadoc

Interfaccia	Metodi	Ubicazione delle descrizioni del metodo 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>I metodi definiti nell'interfaccia IACTLibrary sono disponibili anche sull'interfaccia IActionLibrary.</p>	com.ibm.correlation.IActionLibrary
IExitableActionLibrary	<ul style="list-style-type: none"> • exitRuleSet • exitRuleBlock <p>I metodi definiti nell'interfaccia IACTLibrary e IActionLibrary sono disponibili anche sull'interfaccia IExitableActionLibrary.</p>	com.ibm.correlation.IExitableActionLibrary

Variabile act_location

La variabile act_location è una stringa che identifica l'ubicazione, all'interno della gerarchia di regole di un'espressione.

Dettagli

L'ubicazione è un nome completo che indica la posizione dell'espressione nella gerarchia delle regole. Essa è nel formato *identifier.identifier....*, in cui ogni ricorrenza di *identifier* è uno dei seguenti:

- Il valore dell'attributo name per un elemento XML che è nella rispettiva gerarchia.
- Per gli elementi che ricorrono più volte in un blocco di regole o in una regola e non hanno attributo name: l'elemento XML che contiene l'espressione, seguito da

un numero di indice in parentesi. Questo numero di indice indica la posizione dell'espressione all'interno dell'elemento che contiene. Il contatore per l'assegnazione dei numeri di indice inizia con 0 piuttosto che con 1. Di conseguenza, se un elemento è contenuto in un terzo elemento <action>, ad esempio, il numero di indice viene riportato come `action[2]`.

Questi identificatori sono in ordine discendente da un blocco di regole di livello massimo all'elemento di livello minimo dell'elemento che contiene l'espressione.

Esempio di codifica

Il seguente codice ha accesso alla variabile `act_lib` per ottenere l'ubicazione dell'espressione:

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

Esempi dell'ubicazione che viene restituita dalla variabile

Di seguito sono riportati esempi dell'ubicazione che viene restituita dalla variabile `act_location`

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

Questa espressione è contenuta in:

- Blocco di regole con un valore di attributo name `ruleBlockA`
- La regola con un valore di attributo name `ruleA`
- Il quarto elemento <eventSelector>
- L'elemento <filteringPredicate>

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

Questa espressione è contenuta in:

- Blocco di regole con un valore di attributo name `ruleBlockA`
- La regola con un valore di attributo name `ruleA`
- Elemento <onDetection>
- Sesto elemento <action>

`ruleBlockA.ruleA.variableA.varInitializer`

Questa espressione è contenuta in:

- Blocco di regole con un valore di attributo name `ruleBlockA`
- La regola con un valore di attributo name `ruleA`
- Variabile con valore di attributo name `variableA`
- Elemento <varInitializer>

Variabile `act_nodeName`

La variabile `act_nodeName` è una stringa che identifica il nome completo di un nodo.

Dettagli

In Active Correlation Technology, un nodo è un oggetto nella gerarchia di regole che può essere aggiunto, rimosso o sostituito singolarmente e in modo autonomo all'interno della serie di regole. Questi oggetti possono essere nodi, regole, blocchi di regole, variabili di blocchi di regole e di serie di regole. Poiché un oggetto non può essere elaborato singolarmente e in modo indipendente al di sotto del livello di regola, la variabile di regola non è un nodo.

Il nome nodo completo per una regola con un valore di attributo di nodo rule1 ubicato in un blocco di regole con un valore di attributo name ruleBlockA è ruleBlockA.rule1. Poiché le regole sono organizzate in modo gerarchico all'interno di una serie di regole, utilizzare un punto (.) nel nome nodo completo per denotare un discendente in un nodo di livello inferiore.

Esempio di codifica

Il seguente codice ha accesso alla variabile act_nodeName per ottenere il nome completo di un nodo:

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

Variabile act_threshold

La variabile act_threshold è il valore dell'attributo di soglia degli elementi <computedThreshold> o <eventCountThreshold> di una regola di soglia.

Dettagli

La variabile act_threshold è valida solo all'interno di una regola di soglia.

Se una regola viene definita con una chiave di raggruppamento, le variabili act_eventCount, act_eventList e act_threshold non sono valide all'interno dei seguenti contesti di espressione:

- Azioni attinenti al ciclo di vita
- <filteringPredicate> in <activateOnEvent> o <deactivateOnEvent> in <activationInterval>
- <computedValue>

Ciò in quanto, in questo caso, le variabili di regola si applicano solo a un'istanza della regola e le istanze della regola non esistono al momento in cui queste espressioni vengono eseguite.

Esempio di codifica

Il seguente codice ha accesso alla variabile act_lib per ottenere un valore di soglia definito:

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

Flusso di eventi tramite una serie di regole

Flusso di eventi tramite una serie di regole nell'ordine in cui sono codificati i blocchi di regole e le regole. Quando l'engine dell'Active Correlation Technology riceve un evento, esso determina il tipo di evento e identifica le regole che utilizzano questo tipo di evento per l'attivazione della regola, l'elaborazione dell'evento o la disattivazione della regola.

Come gli eventi vengono utilizzati dalle regole

Ogni regola che utilizza l'evento determina prima se l'evento risponde a tutti i criteri specificati per l'attivazione della regola, l'elaborazione evento o la disattivazione della regola. In questo caso la regola esegue la procedura di seguito riportata.

Per l'attivazione della regola

Le azioni nell'elemento <onActivation> per la regola vengono eseguite se sono codificate.

Per l'elaborazione evento

La regola elabora l'evento. Quando lo schema di regola corrisponde, vengono eseguite le azioni di risposta alla regola, se sono codificate. In alcune situazioni, le azioni di risposta possono avere i comportamenti di seguito riportati.

- L'azione può causare all'evento di ignorare l'elaborazione del blocco o della serie di regole.
- L'azione può inviare un evento nuovo o esistente ad un'altra regola o a un blocco di regole per l'elaborazione.

Per la disattivazione della regola

Le azioni nell'elemento <onDeactivation> per la regola vengono eseguite se sono codificate.

Metodi che possono interessare il flusso degli eventi

Active Correlation Technology fornisce i seguenti metodi su cui si può effettuare una chiamata per influenzare il flusso degli eventi tramite la serie di regole. Questi metodi sono disponibili tramite la variabile `act_lib`.

exitRuleSet

Questo metodo specifica che l'evento corrente non viene elaborato da altre regole aggiuntive nella serie di regole.

exitRuleBlock

Questo metodo specifica che l'evento corrente non viene elaborato da altre regole aggiuntive nel blocco di regole o in qualsiasi blocco di regole che lo contiene. Tuttavia esso viene elaborato da altre regole che sono esterne all'ambito del blocco di regole corrente.

forward

Questo metodo specifica che un evento deve essere inviato ad altre regole e ai blocchi di regola anche se la regola corrente non ha completato la propria elaborazione. Ognuna delle altre regole e dei blocchi di regole elaborano poi l'evento completamente prima di restituire l'evento alla regola che ha chiamato il metodo `forward`.

forwardOnCompletion

Questo metodo specifica che un evento deve essere inviato ad altre regole e ai blocchi di regola dopo che la regola corrente ha completato la propria elaborazione.

Capitolo 3. Panoramica sulla scrittura delle regole

Prima di scrivere le regole per correlare gli eventi, è necessario conoscere e pianificare la correlazione degli eventi e progettare le regole. E' possibile utilizzare il programma di creazione regole dell'Active Correlation Technology per scrivere le regole e compilare la serie di regole.

Il programma di creazione dell'Active Correlation Technology è una GUI per la scrittura delle regole. Esso include la guida in linea. Nel programma di creazione delle regole, il file di serie di regole che ne risulta è un file XML con un tipo file actl.

Active Correlation Technology non fornisce una gestione delle modifiche o un sistema di controllo versione.

Pianificazione per la correlazione eventi

La pianificazione per la correlazione eventi presuppone la comprensione di cosa è la correlazione e di come si realizza nella propria applicazione.

Assicurarsi di comprendere i seguenti concetti:

- Le informazioni in Capitolo 1, "Introduzione", a pagina 3 e Capitolo 2, "Panoramica sul linguaggio delle regole", a pagina 5
- Gli eventi che l'applicazione elabora

Ogni applicazione può elaborare una serie differente di eventi, come descritto nei seguenti esempi:

Esempio di società di assicurazioni

In una società di assicurazioni, gli eventi che tracciano il flusso di lavoro tramite l'elaborazione dei reclami possono essere generati e correlati in maniera tale da determinare se i processi aziendali vengono completati tempestivamente.

Esempio di vendite

In un tipo diverso di attività commerciale, i risultati delle vendite possono essere riepilogati, notificati e confrontati periodicamente, in modo da indicare lo stato degli obiettivi di vendita per un certo periodo di tempo.

Esempio di ambiente IT

In un ambiente IT, un sistema dedicato può generare un evento ogni minuto per indicare se un server di database sta funzionando regolarmente. Le regole di correlazione possono essere scritte per controllare la ricevuta di questi eventi di impulsi e eseguire delle azioni di risposta alla regola se un evento di impulso non viene ricevuto.

E' necessario capire il formato degli eventi che l'applicazione elabora. Active Correlation Technology fornisce le classi e i metodi Java per accedere ai dati all'interno degli eventi che vengono elaborati dal motore Active Correlation Technology. Tuttavia è importante avere una conoscenza di base degli oggetti di evento sottostanti se si desidera utilizzare classi e metodi per accedere o modificare gli eventi quando sono in elaborazione.

Per pianificare la correlazione eventi, seguire questa procedura:

1. Definire gli eventi dall'applicazione da correlare.
2. Definire gli schemi di regola per la correlazione degli eventi.

Uno schema di regola rappresenta una situazione di correlazione eventi e può essere utilizzata per correlare gli eventi che contribuiscono in qualche modo a quella situazione. Riflettere su come gli eventi che l'applicazione elabora sono correlati agli schemi di regola che sono definiti dal linguaggio di regola dell'Active Correlation Technology. Ciò può aiutare a determinare gli schemi di regola da utilizzare.

Utilizzare sempre il modello più appropriato per la condizione di correlazione eventi. Ad esempio per rilevare una regola per una data sequenza di eventi, non scrivere il codice per includere il modello di sequenza nelle azioni di risposta di regola per una regola di filtro. Utilizzare piuttosto il modello di sequenza per creare una regola di sequenza.

3. Identificare i costrutti di ciascuno schema di regola da utilizzare.

Le informazioni che seguono sono un riepilogo dei costrutti primari nel linguaggio di regola, sebbene i dettagli di ciascuno di essi sono univoci per lo schema di regola. Queste informazioni sono organizzate approssimativamente nello stesso modo in cui vengono presentate tramite la GUI del programma di creazione regola:

Caratteristiche

La definizione delle caratteristiche di una regola, compreso il nome regola, la descrizione e lo schema. Per i dettagli, consultare le seguenti sezioni:

- "Elemento collectionRule" a pagina 75
- "Elemento computationRule" a pagina 77
- "Elemento duplicateRule" a pagina 85
- "Elemento filterRule" a pagina 92
- "Elemento sequenceRule" a pagina 107
- "Elemento thresholdRule" a pagina 112
- "Elemento timerRule" a pagina 114

Variabili

La definizione delle variabili di regola, compreso il nome, il tipo, la descrizione e l'espressione di inizializzazione per ciascuna variabile. Per i dettagli, consultare "Elemento variable" a pagina 116.

Selezione evento

La definizione dei criteri che determinano gli eventi che sono accettati per l'elaborazione della regola. Per i dettagli, consultare "Elemento eventSelector" a pagina 89.

Chiave di raggruppamento

La definizione della chiave di raggruppamento che è un metodo per indirizzare una regola in modo da creare un'istanza della regola separata (o una sua copia) per ciascun gruppo di eventi che condivide caratteristiche comuni. Per i dettagli, consultare "Elemento groupingKey" a pagina 93.

Specifiche di schema

La specifica del periodo di tempo durante il quale la regola stateful è in elaborazione per adattarsi al suo schema e la definizione degli aspetti univoci di alcuni schemi di regola. Per i dettagli, consultare "Elemento timeWindow" a pagina 116.

Nella regola di calcolo è inclusa la definizione del calcolo da applicare agli eventi raccolti. Per i dettagli, consultare “Elemento computeFunction” a pagina 81.

Nella regola di soglia è inclusa la definizione del tipo di soglia e le informazioni specifiche per il tipo di soglia. Per i dettagli, consultare le seguenti sezioni:

- “Elemento booleanThreshold” a pagina 75
- “Elemento computedThreshold” a pagina 78
- “Elemento eventCountThreshold” a pagina 86

Risposte della regola

Le definizioni delle azioni da intraprendere quando la regola completa la sua elaborazione.

Per i dettagli, consultare le seguenti sezioni:

- Per le regole di duplicato, filtro, sequenza o soglia: “Elemento onDetection” a pagina 100
- Per le regole di duplicato: “Elemento onNextEvent” a pagina 101
- Per le regole di sequenza o soglia: “Elemento onTimeOut” a pagina 102
- Per le regole di raccolta, calcolo, duplicato o timer: “Elemento onTimeWindowComplete” a pagina 103

Intervallo di attivazione

La definizione di quando una regola è attiva e inattiva. Per i dettagli, consultare “Elemento activationInterval” a pagina 69.

Durata

Le azioni da intraprendere nelle quattro fasi primarie nel ciclo di vita di una regola: caricamento, attivazione, disattivazione e scaricamento. In genere queste azioni non devono essere definite. Per i dettagli, consultare “Elemento lifeCycleActions” a pagina 97.

4. Identificare i metodi e i frammenti Java associati da richiamare all’interno delle espressioni di regola. Piuttosto che scrivere codice Java all’interno delle espressioni di regole, gli scrittori di regola utilizzano i metodi Java per richiamare i moduli esterni. Questi moduli esterni possono essere forniti dall’applicazione che integra Active Correlation Technology o possono essere creati dallo scrittore di regole quando necessario. Devono essere identificati anche i frammenti associati ad ognuno dei metodi Java. Per ulteriori informazioni, consultare “Procedure ottimali per le espressioni di codifica” a pagina 23.

Procedere a “Designazione delle regole da correlare agli eventi”.

Designazione delle regole da correlare agli eventi

Dopo avere pianificato la correlazione eventi, è necessario designare le regole da correlare agli eventi.

Prima di tutto completare “Pianificazione per la correlazione eventi” a pagina 35.

Per designare le regole, seguire questa procedura:

1. Designare l’organizzazione delle regole e i blocchi di regole all’interno della serie di regole.

2. Designare il livello su cui devono essere definite variabili differenti, ad esempio a livello di serie di regole, di blocco di regole o di regola.
3. Designare gli elementi di ogni regola basata sullo schema di regola.
Questa fase utilizza i costrutti di ciascuno schema di regola identificato nella fase di pianificazione.
4. Designare le interazioni tra le regole, specie in relazione all'inoltro di eventi e ignorando l'elaborazione di serie di regole per gli eventi duplicati.
Per ulteriori dettagli, consultare "Flusso di eventi tramite una serie di regole" a pagina 33.
5. Progettare, sviluppare e provare i metodi Java da richiamare all'interno delle espressioni.

Procedere a "Introduzione al programma di creazione di regole".

Introduzione al programma di creazione di regole

Dopo aver progettato le regole per essere correlate agli eventi, è possibile utilizzare il programma di creazione di regole dell'Active Correlation Technology per scrivere le regole.

La procedura che segue è quella principale per scrivere le regole utilizzando il programma di creazione regole.

1. Aprire Eclipse Workbench.
2. Impostare la propria prospettiva in Eclipse Workbench.
3. Impostare le preferenze per Active Correlation Technology o accettare le preferenze predefinite.
4. Creare un progetto in cui memorizzare il file della serie di regole o utilizzare un progetto esistente.
5. Creare un file di serie di regole e memorizzarlo nel progetto scelto.
6. Creare almeno un blocco di regole all'interno di una serie. Inoltre è possibile creare altri blocchi di regole all'interno di una serie ed è possibile creare blocchi di regole all'interno di blocchi di regole.
7. Creare regole in blocchi di regole.
8. Convalidare la serie di regole.
9. Compilare la serie di regole.
10. Aggiornarlo se necessario.

Nelle espressioni all'interno delle regole, è anche possibile includere i frammenti della vista in Eclipse Workbench.

Impostazione di prospettiva in Eclipse Workbench

Prima di tutto impostare la propria prospettiva in Eclipse Workbench. Queste informazioni descrivono come aprire la prospettiva di costruzione regole.

In primo luogo aprire Eclipse Workbench, se non è già aperto.

Sebbene sia possibile utilizzare una prospettiva diversa da quella di costruzione regole, le azioni da eseguire per le diverse attività variano leggermente a seconda della prospettiva scelta.

Per aprire la prospettiva di costruzione regole, seguire questa procedura:

1. Nell'area di lavoro, fare clic su **Finestra** → **Apri prospettiva** → **Altro....**

2. Fare clic su **Costruzione regole** e su **OK**. Viene poi visualizzata la prospettiva di costruzione regole.

Procedere a “Impostazione delle preferenze”.

Impostazione delle preferenze

Prima di creare un file di serie di regole, assicurarsi che le preferenze per Active Correlation Technology siano impostate in Eclipse Workbench. Queste informazioni descrivono come impostare queste preferenze.

In primo luogo aprire Eclipse Workbench, se non è già aperto.

Per impostare le preferenze per Active Correlation Technology:

1. Nel workbench, fare clic su **Finestra** → **Preferenze...**
2. Fare clic su **Active Correlation Technology** e sulla pagina Active Correlation Technology, specificare se aggiungere “act” come prefisso per le regole appena create e per i blocchi di regole nella costruzione regole. “act” *non* viene aggiunto automaticamente come prefisso.
3. Espandere **Active Correlation Technology**. A seconda dell’applicazione, possono essere visualizzate le seguenti altre voci. Fare clic su queste voci per impostare le preferenze associate.

Voce	Preferenza associata
Provider di definizione evento di base comune	E’ possibile specificare i file XML che forniscono la struttura per uno o più tipi di evento (inclusi i nomi e i tipi di attributi che un tipo di evento fornito può contenere) conformi alla specifica CBE (Common Base Event). Questi tipi e attributi di evento sono poi disponibili quando si creano le regole.
Compilatore	<p>E’ possibile specificare le seguenti voci principali :</p> <ul style="list-style-type: none">• Se salvare le serie di regole compilate• Il tipo file per le serie di regole compilate• Il percorso della classe per il codice da utilizzare nella compilazione <p>Come impostazione predefinita, le serie di regole compilate vengono salvate e visualizzate con tipo file acts che indica che si tratta di una serie di regole ordinata.</p>

Procedere a “Creazione di un progetto per la memorizzazione di un file di una serie di regole”.

Creazione di un progetto per la memorizzazione di un file di una serie di regole

In Eclipse Workbench, quando si crea una serie di regole, è necessario specificare il progetto in cui memorizzare il file. Di conseguenza è necessario creare un progetto prima di creare la serie di regole o utilizzare un progetto esistente. Le informazioni che seguono descrivono un modo per creare un progetto in Eclipse Workbench.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Per creare un progetto semplificato in Workbench, seguire questa procedura:

1. Fare clic su **File** → **Nuovo** → **Progetto...**
2. Nella procedura guidata Nuovo progetto, fare clic su **Progetto** → **semplice** e fare clic su **Avanti**.
3. Nel campo **Nome progetto**, immettere un nome univoco per il progetto. Utilizzare inoltre l'ubicazione predefinita per il progetto o sceglierne una differente.
4. Fare clic su **Fine**. Il nuovo oggetto viene poi visualizzato nel riquadro di navigazione della prospettiva di costruzione di regole.

Procedere a "Creazione di una serie di regole".

Creazione di una serie di regole

Queste informazioni descrivono come creare una serie di regole.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Per creare un file di serie di regole, eseguire queste azioni:

1. Fare clic su **File** → **Nuovo** → **File di serie di regole**.
2. Nella procedura guidata Nuovo file di serie di regole, fare clic sul nome della cartella associata al progetto in cui memorizzare il file di serie di regole. E' probabilmente il progetto creato in "Creazione di un progetto per la memorizzazione di un file di una serie di regole" a pagina 39. Il nome della cartella viene poi riportato nel primo campo.
3. Nel campo **Nome file**, immettere un nome per il file di serie di regole. Il tipo file deve essere actl.
4. Fare clic su **Avanti**.
5. Immettere un nome univoco per la serie di regole e una descrizione facoltativa. Se non si desidera utilizzare il valore predefinito per il campo **Codifica XML**, specificare anche lo stile della codifica per il file di serie di regole, che è un file XML.
6. Fare clic su **Fine**. Il file di serie di regole viene riportato nella cartella del progetto. Poiché il file viene convalidato quando viene salvato, accanto al file viene riportata la parola "Convalidato". Viene anche riportato il file di serie di regole.

Procedere a "Creazione di un blocco di regole".

Creazione di un blocco di regole

Queste informazioni descrivono come creare un blocco di regole in una serie di regole o in un altro blocco di regole.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Se si sta creando una serie di regole per la prima volta, è necessario creare almeno un blocco di regole all'interno della serie di regole prima di poter creare le regole. Una volta creato il primo blocco di regole, è possibile creare altri blocchi di regole anche all'interno di blocchi di regole.

Per creare un blocco di regole, eseguire queste azioni:

1. Nel riquadro di navigazione, fare doppio clic sul nome del file di serie di regole da aggiornare. Il file viene poi aperto nella vista Panoramica. Quando viene evidenziata la serie di regole nella vista Panoramica, nell'area editor vengono mostrate le informazioni correnti per la serie di regole.
2. Nella vista Panoramica, fare clic con il tastino destro del mouse sulla serie di regole.
3. Fare clic su **Nuovo secondario** → **Blocco di regole**. Il blocco di regole viene mostrato nella serie di regole della vista Panoramica e le informazioni correnti per il blocco di regole vengono riportate nell'area editor.

E' possibile creare altri blocchi di regole nei modi seguenti:

- Per creare un blocco di regole che è un nodo peer per un blocco di regole esistente, fare clic con il tastino destro del mouse sul blocco di regole esistente e su **Nuovo pari livello** → **Blocco di regole**.
- Per creare un blocco di regole in un blocco di regole esistente, fare clic con il tastino destro del mouse sul blocco di regole esistente e fare clic su **Nuovo secondario** → **Blocco di regole**.

Inoltre, tramite l'editor, è possibile aggiornare le informazioni per ogni serie e blocco di regole. Procedere a "Creazione di una regola".

Creazione di una regola

Queste informazioni descrivono come creare una regola.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Una regola va creata all'interno di un blocco di regole. Per creare una regola, eseguire queste azioni:

1. Nella vista di riquadro, fare clic con il tastino destro del mouse sul blocco di regole da aggiornare.
2. Fare clic su **Nuovo secondario** e il tipo di regola da creare. La regola viene poi mostrata nel blocco di regole della vista di riquadro e le informazioni correnti per la regola vengono riportate nell'area editor.

E' anche possibile aggiungere, nello stesso modo, regole ai blocchi di regole. Inoltre, tramite l'editor, è possibile aggiornare le informazioni per ogni regola. Procedere a "Convalida di una serie di regole".

Convalida di una serie di regole

Queste informazioni descrivono come convalidare una serie di regole prima di compilarla.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Per convalidare una serie di regole, eseguire queste azioni:

1. Nella vista Panoramica, fare clic con il tastino destro del mouse su una regola, una serie di regole o su un blocco di regole.
2. Fare clic su **Convalida serie di regole**. Una volta completata la convalida, viene visualizzata una finestra di messaggi per indicare eventuali problemi. Se la convalida viene completata correttamente, alla destra del nome file per la serie di regole, viene riportata la parola "Convalidata".

Una volta completata senza errori la convalida, procedere a “Compilazione di una serie di regole”.

Compilazione di una serie di regole

Queste informazioni descrivono come compilare una serie di regole.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva del programma di creazione di regole.

Nella vista di navigazione o di riquadro, fare clic con il tastino destro del mouse sulla serie di regole da compilare e fare clic su **Compila serie di regole**. Tutti gli errori di compilazione vengono riportati nella vista Problemi.

Se non vi sono errori di compilazione, la serie di regole compilata viene riportata automaticamente con un tipo di file predefinito `acts`, che indica che si tratta di una serie di regole ordinata. Alla destra del nome file, per la serie di regole, viene riportata la parola “Compilata”.

Aggiornamento di una serie di regole

Queste informazioni descrivono come aggiornare una serie di regole.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

Nella vista di riquadro, fare clic sulla voce (regola, blocco di regole o serie di regole) da aggiornare. Le informazioni correnti per la voce vengono poi riportate nell’area editor ed è possibile utilizzare l’editor per aggiornare queste informazioni.

Per creare una regola o un blocco di regole di nodo peer per un blocco di regole o regola esistenti, eseguire queste azioni:

1. Fare clic con il tastino destro del mouse sul blocco di regole o regola esistenti.
2. Fare clic su **Nuovo pari livello** e sulla voce (blocco di regole o tipo di regola) da aggiungere.

Per creare una regola o un blocco di regole all’interno di un blocco di regole esistente, eseguire queste azioni:

1. Fare clic con il tastino destro del mouse sul blocco di regole esistente.
2. Fare clic su **Nuovo secondario** e sulla voce (blocco di regole o tipo di regola) da aggiungere.

Per accedere ad altre funzioni di aggiornamento nella vista di riquadro, seguire questa procedura:

1. Fare clic con il tastino destro del mouse sulla voce (blocco di regole o regola) da aggiornare.
2. Fare clic sulla voce di menu per la funzione, come **Elimina**, **Copia** o **Incolla**.

Inclusione di frammenti nelle espressioni all’interno delle regole

Nelle espressioni all’interno delle regole, è anche possibile includere i frammenti dalla vista Frammenti di Eclipse Workbench.

In primo luogo aprire Eclipse Workbench, se non è già aperto. Aprire anche la prospettiva di costruzione regole.

La vista dei frammenti è riportata nella prospettiva di costruzione regole. I frammenti sono organizzati in categorie basate sulla loro funzione.

Per includere un frammento dalla vista relativa, eseguire questa procedura:

1. Fare clic su una categoria di frammenti per visualizzarne i nomi nella categoria.
2. Fare clic sul frammento da includere in un'espressione.
3. Trascinare il frammento nel relativo campo di **Espressione**. Il codice viene posizionato sul cursore nel campo **Espressione**. Se il codice richiede l'input dal programma di scrittura delle regole, come il testo specifico o i valori di variabile, viene richiesto di fornire quell'input prima che il codice venga incluso nell'espressione.

Procedere a “Convalida di una serie di regole” a pagina 41.

Parte 2. Riferimenti per il programma di scrittura delle regole

Capitolo 4. Riepilogo sull'organizzazione della serie di regole

Questo riepilogo elenca tutti gli elementi di linguaggio di una serie di regole, un blocco di regole ed ogni tipo di regola. Esso serve come riferimento rapido per la codifica di una serie di regole.

Tabella 10 illustra il significato della notazione che segue ogni elemento di linguaggio. n rappresenta un numero.

Tabella 10. Spiegazione della notazione che definisce il numero di ricorrenze per un elemento di linguaggio

Notazione	Significato
(0, 1)	0 indica che l'elemento di linguaggio è facoltativo. 1 indica che, se è codificata, è consentita solo una ricorrenza.
(0, n)	0 indica che l'elemento di linguaggio è facoltativo. n indica che se è codificato, sono consentite più ricorrenze.
(1, 1)	Il primo 1 indica che l'elemento di linguaggio è obbligatorio. Il secondo 1 indica che è consentita solo una ricorrenza.
(1, n)	1 indica che l'elemento di linguaggio è obbligatorio. n indica che sono consentite più ricorrenze.
(2, n)	2 indica che sono obbligatorie 2 ricorrenze dell'elemento di linguaggio. n indica che sono consentite ulteriori ricorrenze.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Riepilogo della serie di regole

Questo riepilogo elenca gli elementi di linguaggio di una serie di regole.

Elementi della serie di regole

<ruleSet> contiene i seguenti elementi:

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

Riferimenti correlati

“Riepilogo del blocco di regole”

Questo riepilogo elenca gli elementi di linguaggio di un blocco di regole.

Riepilogo del blocco di regole

Questo riepilogo elenca gli elementi di linguaggio di un blocco di regole.

Elementi del blocco di regole

`<ruleBlock>` contiene i seguenti elementi.

Se essi sono codificati, gli elementi `<comment>`, `<import>` e `<variable>` devono essere codificati nell'ordine riportato. Gli elementi rimanenti possono essere codificati in un ordine qualsiasi.

- `<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)

Riferimenti correlati

“Riepilogo di regola Collection”

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di raccolta.

“Riepilogo della regola computation” a pagina 50

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di calcolo.

“Riepilogo regola di duplicato” a pagina 51

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di duplicato.

“Riepilogo di regola di filtro” a pagina 52

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di filtro.

“Riepilogo della regola di sequenza” a pagina 53

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di sequenza.

“Riepilogo della regola di soglia (Threshold)” a pagina 55

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di soglia.

“Riepilogo di regola Timer” a pagina 56

Questo riepilogo elenca tutti gli elementi di linguaggio della regola timer.

Riepilogo di regola Collection

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di raccolta.

Elementi della regola

`<collectionRule>` contiene i seguenti elementi:

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

- <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - I seguenti tre elementi, in qualsiasi ordine (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Uno dei seguenti due elementi (1, 1):

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

Riepilogo della regola computation

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di calcolo.

Elementi della regola

<computationRule> contiene i seguenti elementi:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - I seguenti tre elementi, in qualsiasi ordine (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <computeFunction> (1, 1)
- <timeWindow> (1, 1)
 - Uno dei seguenti due elementi (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Riepilogo regola di duplicato

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di duplicato.

Elementi della regola

<duplicateRule> contiene i seguenti elementi:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - I seguenti tre elementi, in qualsiasi ordine (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Uno dei seguenti due elementi (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onNextEvent> (0, 1)
 - <action> (0, n)
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Riepilogo di regola di filtro

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di filtro.

Elementi della regola

<filterRule> contiene i seguenti elementi:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)

Riepilogo della regola di sequenza

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di sequenza.

Elementi della regola

<sequenceRule> contiene i seguenti elementi:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - I seguenti tre elementi, in qualsiasi ordine (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- <timeWindow> (1, 1)
 - Uno dei seguenti due elementi (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onTimeOut> (0, 1)
 - <action> (0, n)

Riepilogo della regola di soglia (Threshold)

Questo riepilogo elenca tutti gli elementi di linguaggio della regola di soglia.

Elementi della regola

<thresholdRule> contiene i seguenti elementi:

- <comment> (0, 1)
- <variable> (0, n)
 - <comment> (0, 1)
 - <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - I seguenti tre elementi, in qualsiasi ordine (1, n):
 - <attributeAlias>
 - <eventAttribute> (2, n)
 - <attributeName>
 - <computedValue>
- Uno dei seguenti tre elementi (1, 1):
 - <booleanThreshold>
 - <computedThreshold>
 - <eventCountThreshold>
- <timeWindow> (1, 1)
 - Uno dei seguenti due elementi (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onDetection> (0, 1)
 - <action> (0, n)
- <onTimeOut> (0, 1)
 - <action> (0, n)

Riepilogo di regola Timer

Questo riepilogo elenca tutti gli elementi di linguaggio della regola timer.

Elementi della regola

<timerRule> contiene i seguenti elementi:

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

- <comment> (0, 1)
- <varInitializer> (1, 1)
- <activationInterval> (0, 1)
 - <activationTime> (0, 1)
 - <start> (0, 1)
 - Uno dei seguenti tre elementi (1, 1):
 - <dateTime>
 - <whenLoaded>
 - <inactiveWhenLoaded>
 - <stop> (0, 1)
 - Uno dei seguenti tre elementi (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)
 - Uno dei seguenti due elementi (1, 1):
 - <timeInterval>
 - <runUntilDeactivated>
- <onTimeWindowComplete> (0, 1)
 - <action> (0, n)

Capitolo 5. Panoramica sugli elementi di linguaggio

Questa panoramica descrive i dettagli degli elementi di linguaggio nello schema XML per il linguaggio delle regole dell'Active Correlation Technology. Gli elementi di linguaggio vengono elencati in ordine alfabetico e gli attributi che sono disponibili per ciascun elemento sono descritti all'interno della sezione per quell'elemento.

In XML e in altri linguaggi di markup come SGML e HTML, un elemento è un'unità di base che consiste di una tag di inizio, una tag di fine e di attributi associati e rispettivi valori e qualsiasi testo contenuto tra le tag di inizio e di fine. Un attributo è una coppia nome-valore codificato su un elemento per definire una certa caratteristica dell'elemento. Un attributo ha un tipo di dati che identifica il tipo di informazioni fornito nel suo valore (ad esempio, numerico, testuale o booleano).

In XML, uno spazio per i nomi è un URI (uniform resource identifier) che fornisce un nome univoco da associare agli elementi e alle definizioni di tipo in uno schema. L'URI indica quale schema XML contiene la definizione di un elemento. Uno spazio per i nomi viene specificato con una stringa di prefisso seguita da due punti. Lo schema di linguaggio di regole dell'Active Correlation Technology viene definito in tre diversi file e utilizza i seguenti tre spazi per i nomi:

- xsd:** Questo spazio per i nomi indica che l'elemento di linguaggio viene definito nello schema XML standard che è descritto in <http://www.w3.org>.
- br:** Questo spazio per i nomi indica che l'elemento di linguaggio viene definito nello schema di serie di regole dell'Active Correlation Technology che è nel file `ACTparser://www.jar.xsd` nella sottodirectory `com/ibm/correlation/ruleparser/xml/RuleSetBase.xsd`. Ad esempio `br:ruleSet` fa riferimento all'elemento `ruleSet` che è definito nel file `RuleSetBase.xsd`.
- act:** Questo spazio per i nomi indica che l'elemento di linguaggio viene definito nello schema di linguaggio dell'Active Correlation Technology che è nel file `ACTparser.jar` nella sottodirectory `com/ibm/correlation/ruleparser/xml/ACTL.xsd`. Ad esempio `act:ruleSet` fa riferimento all'elemento `ruleSet` che è definito nel file `ACTL.xsd`.

Nello schema di linguaggio della regola, gli elementi di linguaggio sono definiti come elementi o tipi complessi, ad esempio:

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

Nello schema, gli attributi `minOccurs` e `maxOccurs` definiscono rispettivamente il numero minimo e massimo di ricorrenze per un elemento di linguaggio. Tabella 11 descrive il significato di valori differenti per gli attributi `minOccurs` e `maxOccurs`.

Tabella 11. Attributi nello schema che definiscono il numero di ricorrenze per un elemento di linguaggio

Attributo	Valore di attributo	Significato
<code>minOccurs</code>	0	L'elemento di linguaggio è facoltativo.

Tabella 11. Attributi nello schema che definiscono il numero di ricorrenze per un elemento di linguaggio (Continua)

Attributo	Valore di attributo	Significato
minOccurs	1	L'elemento di linguaggio deve ricorrere almeno una volta. 1 è il valore predefinito per l'attributo minOccurs.
minOccurs	2	L'elemento di linguaggio deve ricorrere almeno due volte.
maxOccurs	1	L'elemento di linguaggio non può ricorrere più di una volta. 1 è il valore predefinito per l'attributo maxOccurs.
maxOccurs	unbounded	L'elemento di linguaggio può ricorrere un numero qualunque di volte.

Elemento action

L'elemento <action> contiene un'espressione che definisce un'azione di risposta della regola o un'azione attinente al ciclo di vita.

Dettagli

Fare riferimento a "Variabili" a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<action> ha i seguenti attributi:

Tabella 12. Attributi dell'elemento <action>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes
name	Identifica l'azione. Questo identificativo risulta utile per la risoluzione dei problemi, specialmente se si tratta di un nome univoco tra tutti gli elementi <action> definiti per una specifica risposta della regola o per una specifica azione attinente al ciclo di vita.	xsd:NMTOKEN	No

Contenuto in

<action> è contenuto nei seguenti elementi:

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

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

Contiene

<action> non contiene elementi.

Concetti correlati

“Espressioni” a pagina 20

Un’espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all’engine dell’Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento activateOnEvent

L’elemento <activateOnEvent> definisce gli eventi che possono attivare la regola o, per le regole definite con un elemento <groupingKey>, un’istanza della regola.

Di seguito vengono illustrati tre modi possibili di selezionare gli eventi:

- L’uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L’uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L’uso di un elemento <filteringPredicate> senza elementi <eventType>

Se la regola è inattiva e non è stato codificato alcun elemento <eventType> o <filteringPredicate>, vengono selezionati tutti gli eventi che si verificano.

Se non si codifica un elemento <eventType> le prestazioni possono risentirne in modo negativo.

Presumere il caso in cui si selezionano tutti gli eventi del tipo Audit Failure. E’ possibile utilizzare un predicato di filtro per restringere i criteri di scelta in modo da includere solo gli eventi che presentano un attributo di evento con un certo valore. Codificare, ad esempio, un elemento <eventType> per selezionare tutti gli eventi del tipo Audit Failure e codificare un elemento <filteringPredicate> per selezionare solo quegli eventi che hanno un attributo hostname con il valore MyCriticalSystem.

Attributi

<activateOnEvent> non ha attributi.

Contenuto in

<activateOnEvent> è contenuto nei seguenti elementi:

- <activationInterval>
- <activationByGroupingKey>

Contiene

<activateOnEvent> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 13. Elementi contenuti nell'elemento `<activateOnEvent>`

Elemento	Obbligatorio o facoltativo?
<code><eventType></code>	Facoltativo. Sono consentite zero o più ricorrenze.
<code><filteringPredicate></code>	Facoltativo. E' consentito 0 o 1.
<code><stopAfter></code>	Questo elemento è valido solo quando l'elemento <code><activateOnEvent></code> è contenuto nell'elemento <code><activationByGroupingKey></code> . Facoltativo. E' consentito 0 o 1.

Elemento `activationByGroupingKey`

L'elemento `<activationByGroupingKey>` contiene elementi che specificano gli eventi che possono attivare e disattivare un'istanza della regola definita dall'elemento `<groupingKey>`. Poiché l'elemento `<groupingKey>` non è valido per le regole di filtro e timer, l'elemento `<activationByGroupingKey>` non si applica a queste regole.

Dettagli

La funzione fornita dall'elemento `<activationByGroupingKey>` va utilizzata nelle regole in cui si definisce una chiave di raggruppamento. Essa consente di controllare l'attivazione e la disattivazione di istanze della regola basate sulla chiave di raggruppamento. Quando viene codificato l'elemento `<activationByGroupingKey>`, ogni istanza della regola può essere attivata e disattivata singolarmente in base alle condizioni `<activateOnEvent>` e `<deactivateOnEvent>` all'interno di `<activationByGroupingKey>`.

L'esempio che segue illustra l'uso dell'elemento `<activationByGroupingKey>` all'interno della regola di calcolo.

- La seguente regola di calcolo accetta gli eventi di tipo `StockSharesTraded`. Questi eventi indicano il numero di azioni trattate per diverse aziende.
- La chiave di raggruppamento è l'attributo `stockSymbol` di un evento. Il valore di `stockSymbol` è il nome di una azienda specifica.
- Il valore dell'attributo `sharesTraded` di un evento è il numero di azioni trattate per l'azienda relativa (l'azienda è indicata dal valore dell'attributo `stockSymbol`).
- Per un'azienda specifica, viene creato un prospetto che indica il numero di azioni trattate per quella azienda in un intervallo di 10 minuti. Tuttavia, prima che la regola di calcolo possa creare questo prospetto, deve ricevere un evento del tipo `StartReporting`, con il nome della rispettiva azienda come il valore `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>

```

Attributi

<activationByGroupingKey> non ha attributi.

Contenuto in

<activationByGroupingKey> è contenuto nel seguente elemento:

- <activationInterval>

Contiene

<activationByGroupingKey> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 14. Elementi contenuti nell'elemento <activationByGroupingKey>

Elemento	Obbligatorio o facoltativo?
<activateOnEvent>	Facoltativo. E' consentito 0 o 1.
<deactivateOnEvent>	Facoltativo. E' consentito 0 o 1.

Relazione tra gli elementi <activationInterval> e <activationByGroupingKey>

Gli elementi <activateOnEvent> e <deactivateOnEvent> sono contenuti in entrambi gli elementi:

- <activationInterval>
- <activationByGroupingKey>, che è anche contenuto in <activationInterval>

L'attività della regola può differire in base all'attività della regola corrente e in base alle interazioni tra le definizioni <activateOnEvent> e <deactivateOnEvent> all'interno degli elementi <activationInterval> e <activationByGroupingKey>.

L'esempio che segue illustra come queste definizioni possono interagire.

In questo esempio, viene definita una regola duplicata per eliminare gli eventi dai sistemi che sono in fase di manutenzione e per fornire, alla fine del periodo di manutenzione, un prospetto di riepilogo del numero di eventi che sono stati soppressi.

L'impostazione predefinita prevede che una regola in cui è definita una chiave di raggruppamento consenta l'elaborazione di tutti i valori della chiave di raggruppamento. Di conseguenza quando gli eventi rispondono ai criteri di selezione degli eventi per la regola, tutte le istanze della regola sono attive e pronte per accettare questi eventi in base a qualsiasi valore della chiave di raggruppamento. L'intervallo di attivazione per la regola è lo stesso di quello che viene impostato se la regola non ha una chiave di raggruppamento in quanto vengono elaborati tutti gli eventi che rispondono ai criteri di selezione eventi per la regola.

Nell'esempio che segue, la chiave di raggruppamento è hostname e le definizioni dell'elemento <activationInterval> specificano le seguenti azioni:

1. Attivare tutte le istanze della regola quando viene ricevuto un evento del tipo StartMaintenanceModeAllHosts.
2. Disattivare tutte le istanze della regola dopo 2 ore o quando viene ricevuto un evento del tipo StopMaintenanceModeAllHosts.

```
<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>

```

In alcune situazioni, potrebbe essere necessario controllare quali istanze della regola diventano attive e quando diventano attive. Per queste situazioni, è necessario codificare l'elemento `<activationByGroupingKey>`.

L'esempio che segue estende l'esempio precedente ed illustra come sia possibile utilizzare il valore della chiave di raggruppamento per selezionare le istanze della regola consentite per l'elaborazione. Le definizioni all'interno dell'elemento `<activationByGroupingKey>` specificano le seguenti azioni:

1. Consentire l'elaborazione di istanze della regola per nomi host specifici quando si ricevono eventi del tipo `StartMaintenanceMode` per quei nomi host.
2. Disattivare tutte le istanze della regola dopo 2 ore o quando viene ricevuto un evento del tipo `StopMaintenanceMode` per il relativo nome host.

```

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

```

Le istruzioni che seguono sono di riepilogo a ciò che si verifica quando viene codificato l'elemento `<activationByGroupingKey>`:

- Quando l'elemento `<activateOnEvent>` viene codificato all'interno dell'elemento `<activationByGroupingKey>`, è consentita l'elaborazione solo degli eventi che condividono il valore della chiave di raggruppamento dell'evento che ha risposto alla condizione `<activationByGroupingKey> <activateOnEvent>`.
- Quando l'elemento `<deactivateOnEvent>` viene codificato all'interno dell'elemento `<activationByGroupingKey>`, *non* è consentita l'elaborazione degli eventi che condividono il valore della chiave di raggruppamento dell'evento che ha risposto alla condizione `<activationByGroupingKey> <deactivateOnEvent>`.

L'effetto delle diverse definizioni di attivazione e disattivazione sullo stato della regola

Tabella 15 a pagina 66 e Tabella 16 a pagina 68 mostra come lo stato di una regola sia influenzato dalle diverse definizioni di attivazione e disattivazione. In queste tabelle, vengono utilizzate le seguenti convenzioni:

- *A* è un evento di attivazione.
- Nella notazione "*A*[*x*]", *x* rappresenta il valore della chiave di raggruppamento. Ad esempio, *A*[1] è un evento di attivazione con un valore della chiave di raggruppamento equivalente a 1.
- *D* è un evento di disattivazione.
- Nella notazione "*D*[*x*]", *x* rappresenta il valore della chiave di raggruppamento. Ad esempio, *D*[1] è un evento di disattivazione con un valore della chiave di raggruppamento equivalente a 1.

Tabella 15. Modifiche allo stato della regola basate su definizioni di attivazione differenti

Stato della regola iniziale	Lo stato della regola è potenzialmente influenzato da	Stato della regola finale
Inattivo	Ora definita in <activationInterval> <activationTime> <start>	1. La regola è attivata. 2. Le azioni <onActivation> sono in esecuzione.
	Metodo activate()	
	Evento A, definito all'interno di <activationInterval> <activateOnEvent>	3. Sono consentiti tutti i valori della chiave di raggruppamento.
	Evento A[1], definito in <activationByGroupingKey> <activateOnEvent> (senza <stopAfter>)	1. La regola è attivata. 2. Le azioni <onActivation> sono in esecuzione. 3. E' consentito solo il valore 1 della chiave di raggruppamento. Quando lo schema della regola è abbinato a questa istanza della regola, il valore 1 della chiave di raggruppamento non è più consentito.
	Evento A[2], definito in <activateOnEvent> (con <stopAfter>)	1. La regola è attivata. 2. Le azioni <onActivation> sono in esecuzione. 3. Solo il valore 2 della chiave di raggruppamento è consentito ed è consentito solo per la durata specificata dall'elemento <stopAfter>. Lo schema della regola per questa istanza della regola può essere abbinato più volte per tutta la durata.
<ul style="list-style-type: none"> Attivo Consentiti tutti i valori della chiave di raggruppamento 	Ora definita in <activationInterval> <activationTime> <start>	Non si sono verificate modifiche in questo stato della regola. E' uguale allo stato della regola di avvio.
	Metodo activate()	
	Evento A, definito all'interno di <activationInterval> <activateOnEvent>	
	Evento A[1], definito in <activationByGroupingKey> <activateOnEvent> (senza <stopAfter>)	
	Evento A[2], definito in <activateOnEvent> (con <stopAfter>)	

Tabella 15. Modifiche allo stato della regola basate su definizioni di attivazione differenti (Continua)

Stato della regola iniziale	Lo stato della regola è potenzialmente influenzato da	Stato della regola finale
<ul style="list-style-type: none"> Attivo Consentiti solo valori della chiave di raggruppamento che hanno attivato istanze della regola basate sulle definizioni <code><activationByGroupingKey></code> <code><activateOnEvent></code> 	Ora definita in <code><activationInterval></code> <code><activationTime></code> <code><start></code>	Non si sono verificate modifiche in questo stato della regola. E' uguale allo stato della regola di avvio.
	Metodo <code>activate()</code>	
	Evento <i>A</i> , definito all'interno di <code><activationInterval></code> <code><activateOnEvent></code>	Sono consentiti tutti i valori della chiave di raggruppamento.
	Evento <i>A</i> [1], definito in <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>senza</i> <code><stopAfter></code>)	<ul style="list-style-type: none"> Il valore della chiave 1 di raggruppamento è ora consentito oltre ai valori della chiave di raggruppamento precedentemente consentiti. Quando lo schema della regola è abbinato a questa istanza della regola, il valore 1 della chiave di raggruppamento non è più consentito.
<ul style="list-style-type: none"> Attivo Consentiti tutti i valori della chiave di raggruppamento tranne quelli non consentiti in base alle definizioni <code><activationByGroupingKey></code> <code><deactivateOnEvent></code> 	Ora definita in <code><activationInterval></code> <code><activationTime></code> <code><start></code>	Non si sono verificate modifiche in questo stato della regola. E' uguale allo stato della regola di avvio.
	Metodo <code>activate()</code>	
	Evento <i>A</i> , definito all'interno di <code><activationInterval></code> <code><activateOnEvent></code>	Sono consentiti tutti i valori della chiave di raggruppamento.
	Evento <i>A</i> [1], definito in <code><activationByGroupingKey></code> <code><activateOnEvent></code> (<i>senza</i> <code><stopAfter></code>)	Il valore 1 della chiave di raggruppamento è ora consentito oltre ai valori della chiave di raggruppamento precedentemente consentiti.
	Evento <i>A</i> [2], definito in <code><activateOnEvent></code> (<i>con</i> <code><stopAfter></code>)	Il valore 2 della chiave di raggruppamento è ora consentito oltre ai valori della chiave di raggruppamento precedentemente consentiti.

Tabella 16. Modifiche allo stato della regola in base a differenti definizioni di disattivazione

Stato della regola iniziale	Lo stato della regola è potenzialmente influenzato da	Stato della regola finale
Inattivo	Ora definita in <activationInterval> <activationTime> <stop>	Non si sono verificate modifiche in questo stato della regola. E' uguale allo stato della regola di avvio.
	Metodo deactivate()	
	Evento <i>D</i> , definito in <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definito in <activationByGroupingKey> <deactivateOnEvent>	
	La durata definita in <activationByGroupingKey> <activateOnEvent> <stopAfter> termina per evento <i>A</i> [2]	
<ul style="list-style-type: none"> Attivo Consentiti tutti i valori della chiave di raggruppamento 	Ora definita in <activationInterval> <activationTime> <stop>	<ol style="list-style-type: none"> Tutte le istanze della regola vengono disattivate. Le azioni <onDeactivation> sono in esecuzione. La regola è disattivata.
	Metodo deactivate()	
	Evento <i>D</i> , definito in <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definito in <activationByGroupingKey> <deactivateOnEvent>	<ul style="list-style-type: none"> Il valore 1 della chiave di raggruppamento non è più consentito. Se è attiva l'istanza della regola con valore di raggruppamento equivalente a 1, essa viene disattivata.
	La durata definita in <activationByGroupingKey> <activateOnEvent> <stopAfter> termina per evento <i>A</i> [2]	L'istanza della regola con valore di raggruppamento equivalente a 2 è disattivata.
<ul style="list-style-type: none"> Attivo Consentiti solo valori della chiave di raggruppamento che hanno attivato istanze della regola basate sulle definizioni <activationByGroupingKey> <activateOnEvent> 	Ora definita in <activationInterval> <activationTime> <stop>	<ol style="list-style-type: none"> Tutte le istanze della regola vengono disattivate. Le azioni <onDeactivation> sono in esecuzione. La regola è disattivata.
	Metodo deactivate()	
	Evento <i>D</i> , definito in <activationInterval> <deactivateOnEvent>	
	Evento <i>D</i> [1], definito in <activationByGroupingKey> <deactivateOnEvent>	<ul style="list-style-type: none"> Il valore 1 della chiave di raggruppamento non è più consentito. Se è attiva l'istanza della regola con valore di raggruppamento equivalente a 1, essa viene disattivata.
	La durata definita in <activationByGroupingKey> <activateOnEvent> <stopAfter> termina per evento <i>A</i> [2]	<ul style="list-style-type: none"> Il valore 2 della chiave di raggruppamento non è più consentito. L'istanza della regola con valore di raggruppamento 2 viene disattivata.

Tabella 16. Modifiche allo stato della regola in base a differenti definizioni di disattivazione (Continua)

Stato della regola iniziale	Lo stato della regola è potenzialmente influenzato da	Stato della regola finale
<ul style="list-style-type: none"> Attivo Consentiti tutti i valori della chiave di raggruppamento tranne quelli non consentiti in base alle definizioni <code><activationByGroupingKey></code> <code><deactivateOnEvent></code> 	Ora definita in <code><activationInterval></code> <code><activationTime></code> <code><stop></code>	<ol style="list-style-type: none"> Tutte le istanze della regola vengono disattivate. Le azioni <code><onDeactivation></code> sono in esecuzione. La regola è disattivata.
	Metodo <code>deactivate()</code>	
	Evento <i>D</i> , definito in <code><activationInterval></code> <code><deactivateOnEvent></code>	
	Evento <i>D</i> [1], definito in <code><activationByGroupingKey></code> <code><deactivateOnEvent></code>	<ul style="list-style-type: none"> Il valore 1 della chiave di raggruppamento non è più consentito. Se è attiva l'istanza della regola con valore di raggruppamento equivalente a 1, essa viene disattivata.
	La durata definita in <code><activationByGroupingKey></code> <code><activateOnEvent></code> <code><stopAfter></code> termina per evento <i>A</i> [2]	L'istanza della regola con valore di raggruppamento equivalente a 2 è disattivata.

Elemento `activationInterval`

L'elemento `<activationInterval>` contiene elementi che stabiliscono quando una regola è attiva e inattiva.

Dettagli

Una regola può essere attivata o disattivata in una fase definita o da un evento specifico.

Se si specifica che una regola deve essere attivata o disattivata, in una fase definita e da un evento specifico, la regola viene attivata o disattivata in base a ciò che si verifica per primo, la fase o la ricezione dell'evento. Tuttavia, in questo caso, la regola potrebbe essere attivata o disattivata da molti eventi per tutta la durata. Ad esempio una regola potrebbe essere attivata da un evento, disattivata, attivata in una fase definita, disattivata di nuovo e attivata da un altro evento.

In un ambiente aziendale, è possibile attivare una regola quando viene ricevuto un evento che indica che gli uffici della Borsa sono aperti. In un ambiente IT, è possibile avviare una finestra di gestione alle 06:00 del 29 Ottobre 2005 e chiuderla in una delle seguenti circostanze:

- 11:30 del 30 ottobre 2005
- Quando viene ricevuto un evento che indica che l'attività di manutenzione è completata

Attributi

`<activationInterval>` non ha attributi.

Contenuto in

`<activationInterval>` è contenuto nei seguenti elementi:

- `<collectionRule>`

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

Contiene

<activationInterval> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 17. Elementi contenuti nell'elemento <activationInterval>

Elemento	Obbligatorio o facoltativo?
<activationTime>	Facoltativo. E' consentito 0 o 1.
<activateOnEvent>	Facoltativo. E' consentito 0 o 1.
<deactivateOnEvent>	Facoltativo. E' consentito 0 o 1.
<activationByGroupingKey>	Facoltativo. E' consentito 0 o 1.

Relazioni tra gli elementi contenuti

Gli elementi <start> e <stop> che sono contenuti all'interno dell'elemento <activationTime> costituiscono un metodo statico di attivazione e disattivazione di una regola. Tramite questi elementi, viene attivata o disattivata una regola in una fase definita. Al contrario, gli elementi <activateOnEvent> e <deactivateOnEvent> sono un metodo dinamico di attivazione e disattivazione di una regola. Tramite questi elementi, viene attivata o disattivata una regola se si verifica un certo evento. Ad esempio, viene attivata una regola da qualsiasi evento che risponde ai criteri definiti per l'elemento <activateOnEvent>, se la regola non è già attiva. Ad esempio, viene disattivata una regola da qualsiasi evento che risponde ai criteri definiti per l'elemento <deactivateOnEvent>, se la regola non è già inattiva. Di conseguenza certi eventi possono modificare la definizione statica di quando è attivata o disattivata una regola.

Tabella 18 a pagina 71 descrive come e quando una regola viene attivata o disattivata in base a certe combinazioni in cui è possibile codificare i seguenti elementi:

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

In Tabella 18 a pagina 71, X rappresenta il nome di un evento che attiva la regola e Y il nome di un evento che disattiva la regola.

Se l'elemento <start> non viene per niente codificato, l'ora di avvio predefinita è la stessa di quella definita dall'elemento <whenLoaded>.

Se l'elemento <stop> non viene per niente codificato, l'ora di arresto predefinita è la stessa di quella definita dall'elemento <never>.

Tabella 18. L'attività della regola basata sulla codifica di combinazioni differenti degli elementi contenuti in <activationInterval>

<activationTime>		<activateOnEvent>	<deactivateOnEvent>	Attività della regola
<start>	<stop>			
<whenLoaded>	<never>			La regola è attiva quando viene caricata e rimane attiva mentre l'engine dell'Active Correlation Technology è in esecuzione.
<whenLoaded>	<never>		Y	La regola è attiva quando viene caricata. L'evento Y disattiva la regola.
<whenLoaded>	<never>	X	Y	La regola è attiva quando viene caricata. L'evento Y disattiva la regola e l'evento X la riattiva. Questa disattivazione e riattivazione può verificarsi più volte.
<whenLoaded>	<after>			La regola è attiva quando viene caricata e viene disattivata dopo un intervallo di tempo specificato.
<whenLoaded>	<dateTime>			La regola è attiva quando viene caricata e disattivata ad una data e ora specificate.
<inactiveWhenLoaded>	<never>	X		La regola è inattiva quando viene caricata. L'evento X attiva la regola e la regola rimane attiva mentre l'engine dell'Active Correlation Technology è in esecuzione.
<inactiveWhenLoaded>	<never>	X	Y	La regola è inattiva quando viene caricata. L'evento X attiva la regola e l'evento Y la disattiva. Questa attivazione e disattivazione può verificarsi più volte.
<dateTime>	<dateTime>			La regola viene attivata e viene disattivata in una certa data e ora.
<dateTime>	<dateTime>	X	Y	La regola viene attivata e viene disattivata dopo una certa data e ora. L'evento X attiva la regola e l'evento Y la disattiva. L'evento X e Y possono attivare e disattivare la regola più volte.
<dateTime>	<never>			La regola viene attivata in una data e ora specificate e rimane attiva mentre l'engine dell'Active Correlation Technology è in esecuzione.
<dateTime>	<never>		Y	La regola viene attivata in una data e ora specificate. L'evento Y disattiva la regola.
<dateTime>	<never>	X	Y	La regola viene attivata in una data e ora specificate. L'evento Y disattiva la regola e l'evento X la riattiva. Questa disattivazione e riattivazione può verificarsi più volte.
<dateTime>	<after>			La regola viene attivata in una data ed ora specificate e disattivata dopo un intervallo di tempo specificato.
<dateTime>	<after>	X	Y	La regola viene attivata e viene disattivata dopo una certa data e ora. L'evento X attiva la regola e l'evento Y la disattiva. Questa attivazione e disattivazione può verificarsi più volte.

Elemento activationTime

L'elemento <activationTime> definisce i momenti in cui è attivata o disattivata una regola.

Attributi

<activationTime> non ha attributi.

Contenuto in

<activationTime> è contenuto nel seguente elemento:

- <activationInterval>

Contiene

<activationTime> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 19. Elementi contenuti nell'elemento <activationTime>

Elemento	Obbligatorio o facoltativo?
<start>	Facoltativo. E' consentito 0 o 1.
<stop>	Facoltativo. E' consentito 0 o 1.

Elemento after

L'elemento <after> specifica la durata del tempo in cui una regola rimane attiva dopo la relativa attivazione. Dopo questo periodo, la regola deve essere disattivata.

Attributi

<after> ha i seguenti attributi:

Tabella 20. Attributi dell'elemento <after>

Nome	Descrizione	Tipo dati	Obbligatorio
durata	Specifica la quantità di tempo per la durata. Il tipo di dati di questo attributo dipende dal valore dell'attributo unit.	<ul style="list-style-type: none">Se il valore dell'attributo unit è ISO-8601, il tipo di dati è xsd:duration.Se il valore dell'attributo unit è milliseconds, il tipo di dati è xsd:positiveInteger.	Yes
unit	Specifica l'unità di tempo da utilizzare. I valori validi per questo attributo sono: <ul style="list-style-type: none">ISO-8601milliseconds	xsd:string	Yes

Uso dello standard ISO 8601 per la durata di tempo

La codifica ISO-8601 come valore per l'attributo unit indica che il valore dell'attributo di durata è codificato in base allo standard ISO 8601 per specificare una durata di tempo come una stringa. La specifica del tipo di dati dello schema standard XML utilizza ISO 8601 per fornire un tipo di dati chiamato duration. Questo tipo di dati viene descritto in dettaglio su <http://www.w3.org/TR/xmlschema-2/#duration>.

Il formato per il tipo di dati duration nello schema standard XML è la seguente stringa:

PnYnMnDTnHnMnS

- P è il carattere che dà sempre inizio alla stringa.

- *nY* rappresenta il numero di anni. Un anno è lo stesso di 365 giorni. Di conseguenza la codifica 1Y è uguale alla codifica 365D.
- *nM* rappresenta il numero di mesi. Un mese è lo stesso di 30 giorni. Di conseguenza la codifica 1M è uguale alla codifica 30D.
- *nD* rappresenta il numero di giorni.
- T è il separatore che separa le unità di giorni (anni, mesi e giorni) dalle unità di tempo (ore, minuti e secondi). Le unità seguono sempre T.
- *nH* rappresenta il numero di ore.
- *nM* rappresenta il numero di minuti.
- *nS* rappresenta il numero di secondi.

Di seguito sono riportati esempi del formato:

- P5DT12H è 5,5 giorni.
- PT59M59S è 59 minuti e 59 secondi.
- P1M è 1 mese.

Contenuto in

<after> è contenuto nel seguente elemento:

- <stop>

Contiene

<after> non contiene elementi.

Elemento attributeAlias

L'elemento <attributeAlias> fornisce un nome alias per associare gli attributi di evento che hanno lo stesso significato ma nomi diversi in eventi diversi. Ad esempio, tre eventi differenti possono utilizzare questi tre nomi diversi per un attributo di evento che indica il nome del sistema che origina l'evento: host, nomehost e origine. L'elemento <attributeAlias> contiene gli elementi <eventAttribute> che descrivono i singoli attributi di evento che devono essere associati come un attributo di evento per la chiave di raggruppamento.

Dettagli

L'elemento <attributeAlias> e il relativo attributo aliasName sono validi solo nel contesto di una chiave di raggruppamento. Questo elemento e il relativo attributo possono essere indicati in qualsiasi espressione, compreso un'espressione all'interno dell'elemento <computedValue>.

Attributi

<attributeAlias> ha il seguente attributo:

Tabella 21. Attributi dell'elemento <attributeAlias>

Nome	Descrizione	Tipo dati	Obbligatorio
aliasName	Definisce il nome per gli attributi di evento descritti negli elementi <eventAttribute> e per quelli che devono essere associati come un attributo di evento per la chiave di raggruppamento. Questo nome deve essere univoco all'interno della regola.	xsd:NMTOKEN	Yes

Contenuto in

<attributeAlias> è contenuto nel seguente elemento:

- <groupingKey>

Contiene

<attributeAlias> contiene il seguente elemento:

Tabella 22. Elementi contenuti nell'elemento <attributeAlias>

Elemento	Obbligatorio o facoltativo?
<eventAttribute>	Sono richieste 2 ricorrenze di questo elemento. Sono consentite ulteriori ricorrenze.

Elemento attributeName

L'elemento <attributeName> contiene il nome di un attributo di un evento specifico che è parte della chiave di raggruppamento. Questo nome deve corrispondere al nome utilizzato nella chiamata del metodo getAttribute per la variabile act_event.

Attributi

<attributeName> non ha attributi.

Contenuto in

<attributeName> è contenuto nel seguente elemento:

- <groupingKey>

Contiene

<attributeName> non contiene elementi.

Elemento booleanThreshold

L'elemento <booleanThreshold> è valido solo per la regola di soglia. Esso contiene un'espressione chiamata ogni volta che viene ricevuto un evento. L'espressione calcola o confronta il valore di soglia in base all'evento corrente e in base a tutti gli altri eventi che sono stati accettati dalla regola. L'espressione restituisce un valore booleano true o false per indicare se la soglia è stata raggiunta.

Dettagli

Fare riferimento a “Variabili” a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<booleanThreshold> ha il seguente attributo:

Tabella 23. Attributi dell'elemento <booleanThreshold>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes

Contenuto in

<booleanThreshold> è contenuto nel seguente elemento:

- <thresholdRule>

Contiene

<booleanThreshold> non contiene elementi.

Concetti correlati

“Espressioni” a pagina 20

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento collectionRule

L'elemento <collectionRule> definisce una regola in base allo schema di raccolta.

Attributi

<collectionRule> contiene i seguenti attributi:

Tabella 24. Attributi dell'elemento <collectionRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No

Contenuto in

<collectionRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<collectionRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 25. Elementi contenuti nell'elemento <collectionRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<eventSelector>	Facoltativo. E' consentito 0 o 1.
<groupingKey>	Facoltativo. E' consentito 0 o 1.
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onTimeWindowComplete>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema di raccolta" a pagina 11

La regola di raccolta viene definita dallo schema di raccolta. Essa raccoglie un gruppo di eventi selezionati in un intervallo di tempo. Essa è una regola stateful.

Elemento comment

L'elemento <comment> può contenere una descrizione della funzione e della finalità della sua serie di regole, del blocco di regole, della regola o della variabile.

Attributi

<comment> non ha attributi.

Contenuto in

<comment> è contenuto nei seguenti elementi:

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

Contiene

<comment> non contiene elementi.

Elemento computationRule

L'elemento <computationRule> definisce una regola in base allo schema di calcolo.

Attributi

<computationRule> contiene i seguenti attributi:

Tabella 26. Attributi dell'elemento <computationRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No

Contenuto in

<computationRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<computationRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 27. Elementi contenuti nell'elemento <computationRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<eventSelector>	Facoltativo. E' consentito 0 o 1.
<groupingKey>	Facoltativo. E' consentito 0 o 1.
<computeFunction>	Obbligatorio. E' consentita solo una ricorrenza.
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onTimeWindowComplete>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema di calcolo" a pagina 12

Una regola di calcolo viene definita da uno schema di calcolo. Essa applica un calcolo (tramite un'espressione) agli eventi raccolti in quanto ogni evento viene ricevuto in un intervallo di tempo. Essa è una regola stateful.

Elemento computedThreshold

L'elemento <computedThreshold> è valido solo per la regola di soglia. Esso contiene un'espressione chiamata ogni volta che viene ricevuto un evento e che calcola il valore di soglia in base all'evento corrente e su qualsiasi altro evento che risponde ai criteri di selezione evento per la regola. L'espressione restituisce un valore di soglia da memorizzare in una variabile definita per la regola. La regola utilizza poi il valore di soglia calcolato da confrontare in base al valore di soglia definito.

Dettagli

Fare riferimento a "Variabili" a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<computedThreshold> contiene i seguenti attributi:

Tabella 28. Attributi dell'elemento <computedThreshold>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes
soglia	Definisce il valore di soglia che deve essere soddisfatto. Questo valore di soglia definito deve essere una rappresentazione di stringa di un valore numerico che può essere convertito su un tipo di dati valido per la variabile della regola.	xsd:string	Yes
assignTo	Identifica il nome della variabile che mantiene il valore di soglia calcolato che viene restituito da questa espressione. Questa variabile deve già essere definita per la regola (al livello di serie di regole, di blocco di regole o di regole) utilizzando l'elemento <variable>. Essa deve essere definita come uno dei seguenti tipi di dati numerici: <ul style="list-style-type: none"> • java.lang.Double • java.lang.Float • java.lang.Integer • java.lang.Long • java.lang.String Se la variabile viene definita al livello della serie di regole o di blocco di regole, essa non viene nuovamente inizializzata dopo che lo schema di regola viene adattato.	xsd:NMTOKEN	Yes
thresholdComparison	Definisce l'operatore per confrontare il valore di soglia calcolato con il valore di soglia definito. I valori validi per questo operatore sono: <ul style="list-style-type: none"> • lessThan • lessThanOrEqualTo • greaterThan • greaterThanOrEqualTo 	xsd:string	Yes

Contenuto in

<computedThreshold> è contenuto nel seguente elemento:

- <thresholdRule>

Contiene

<computedThreshold> non contiene elementi.

Concetti correlati

“Espressioni” a pagina 20

Un’espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all’engine dell’Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento computedValue

L’elemento <computedValue> contiene un’espressione che viene eseguita quando la regola riceve un evento per creare un valore di stringa basato sul valore di uno o più attributi dell’evento. Questo valore di stringa può quindi essere utilizzato nella chiave di raggruppamento.

Dettagli

Talvolta un programma di scrittura di regole può utilizzare voci come le seguenti nella chiave di raggruppamento:

- Una sottostringa di un valore di attributo di evento. Ad esempio, se un valore di attributo di evento contiene un indirizzo IP integrato, l’espressione nell’elemento <computedValue> può estrarre quell’indirizzo IP come valore univoco da utilizzare nella chiave di raggruppamento.
- Le sottostringhe di valori di diversi attributi di evento. Ad esempio, se un valore di attributo di evento contiene un indirizzo IP integrato, l’espressione nell’elemento <computedValue> può estrarre le sottostringhe e combinarle per creare un valore univoco da utilizzare nella chiave di raggruppamento.

Se l’espressione nell’elemento <computedValue> restituisce un valore null, la regola tratta questo valore null come un valore di attributo mancante.

Fare riferimento a “Variabili” a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L’uso di certe variabili dipende dal contesto dell’espressione.

Attributi

<computedValue> ha il seguente attributo:

Tabella 29. Attributi dell'elemento <computedValue>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes

Contenuto in

<computedValue> è contenuto nel seguente elemento:

- <groupingKey>

Contiene

<computedValue> non contiene elementi.

Concetti correlati

“Espressioni” a pagina 20

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento computeFunction

L'elemento <computeFunction> è valido solo per la regola di calcolo. Esso contiene un'espressione chiamata ogni volta che viene ricevuto un evento e che restituisce un valore da memorizzare in una variabile definita per la regola. Il valore restituito da questa espressione deve corrispondere al tipo di dati della variabile che è definita nell'attributo assignTo dell'elemento <computeFunction>.

Dettagli

Fare riferimento a “Variabili” a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<computeFunction> contiene i seguenti attributi:

Tabella 30. Attributi dell'elemento <computeFunction>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes
assignTo	Identifica il nome della variabile che mantiene il valore restituito da questa espressione. Questa variabile deve già essere definita per la regola (al livello di serie di regole, di blocco di regole o di regole) utilizzando l'elemento <variable>. Se la variabile viene definita al livello della serie di regole o di blocco di regole, essa non viene nuovamente inizializzata dopo che lo schema della regola viene abbinato.	xsd:NMTOKEN	Yes

Contenuto in

<computeFunction> è contenuto nel seguente elemento:

- <computationRule>

Contiene

<computeFunction> non contiene elementi.

Concetti correlati

"Espressioni" a pagina 20

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento dateTime

L'elemento <dateTime> specifica la data e l'ora in cui è attivata o disattivata una regola. Tuttavia la regola viene attivata o disattivata solo se è stata caricata nell'engine dell'Active Correlation Technology prima di quell'ora specificata.

Dettagli

Se la regola non viene caricata nell'engine dell'Active Correlation Technology prima dell'ora specificata per l'attivazione, la regola non viene attivata. Se la regola non viene caricata in un engine dell'Active Correlation Technology in esecuzione prima dell'ora specificata per la disattivazione, la regola viene impostata sullo stato definito dall'elemento <start> e non è mai disattivata dall'elemento <stop>.

Il contenuto dell'elemento <dateTime> deve essere una stringa che segue il formato per il tipo di dati dateTime nello schema XML standard. Ad esempio dateTime consiste di sequenze di lunghezza finita di caratteri nel seguente formato:

aaaa '-' mm '-' gg '0' hh ':' mm ':' ss ('.' s+)? (ffffff)?

- *aaaa* rappresenta l'anno. Non sono consentiti più di quattro cifre zero all'inizio, nonché 0000.
- *'-'*s sono i separatori tra le parti della data.
- Il primo *mm* è un numero a due cifre che rappresenta il mese a partire da 01.
- *gg* è un numero a due cifre che rappresenta il giorno del mese a partire da 01.
- *T* è un separatore che indica che segue l'ora del giorno.
- *hh* è un numero a due cifre che rappresenta l'ora del giorno, nel formato 24 ore, a partire da 00 e che termina con 23.
- *:* è un separatore tra le parti dell'ora.
- Il secondo *mm* è un numero a due cifre che rappresenta il minuto a partire da 00 e che termina con 59.
- *ss* è un numero a due cifre che rappresenta i secondi a partire da 00 e che termina con 59.
- *'.' s+*, se presente, rappresenta i secondi frazionali.
- *ffffff*, se presente, è il fuso orario. Il fuso orario comprende sequenze di caratteri di lunghezza finita nel formato *(('+' | '-') hh ':' mm) | 'F'*, in cui:
 - *'+'*, se presente, rappresenta una durata non negativa e *'-'* non deve essere presente.
 - *'-'*, se presente, rappresenta una durata non positiva e *'+'* non deve essere presente.
 - *hh* è un numero a due cifre che rappresenta le ore a partire da 00 e che termina con 14.
 - *mm* è un numero a due cifre che rappresenta i minuti a partire da 00 e che termina con 59. Tuttavia se il valore delle ore è 14, il valore dei minuti deve essere 00
 - *F* è abbreviazione di UTC (+00:00 o -00:00) e, in questo caso, non devono essere presenti altri elementi di fuso orario.

Di seguito sono riportati due esempi del contenuto dell'elemento <dateTime>:

- 2005-06-01T13:05:06.07 è 1 giugno, 2005 a 6 secondi e 7 centesimi di secondo dopo le 13:05 ora locale.
- 2005-06-01T13:05:06.07 è 1 giugno, 2005 a 6 secondi e 7 centesimi di secondo dopo le 13:05 ora UTC che sarebbe 1 giugno 2005 a 6 secondi e 7 centesimi di secondo dopo le 9:05 EDT (o 2005-06-01T09:05:06.07-04:00).

Attributi

<dateTime> non ha attributi.

Contenuto in

<dateTime> è contenuto nei seguenti elementi:

- <start>
- <stop>

Contiene

<dateTime> non contiene elementi.

Elemento deactivateOnEvent

L'elemento <deactivateOnEvent> definisce gli eventi che possono disattivare la regola o, per le regole definite con un elemento <groupingKey>, un'istanza della regola.

Di seguito vengono illustrati tre modi possibili di selezionare gli eventi:

- L'uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L'uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L'uso di un elemento <filteringPredicate> senza elementi <eventType>

Se la regola è attiva e non è stato codificato alcun elemento <eventType> o <filteringPredicate>, vengono selezionati tutti gli eventi che si verificano.

Se non si codifica un elemento <eventType> le prestazioni possono risentirne in modo negativo.

Presumere il caso in cui si selezionano tutti gli eventi del tipo Audit Failure. E' possibile utilizzare un predicato di filtro per restringere i criteri di scelta in modo da includere solo gli eventi che presentano un attributo di evento con un certo valore. Codificare, ad esempio, un elemento <eventType> per selezionare tutti gli eventi del tipo Audit Failure e codificare un elemento <filteringPredicate> per selezionare solo quegli eventi che hanno un attributo hostname con il valore MyCriticalSystem.

Attributi

<deactivateOnEvent> non ha attributi.

Contenuto in

<deactivateOnEvent> è contenuto nei seguenti elementi:

- <activationInterval>
- <activationByGroupingKey>

Contiene

<deactivateOnEvent> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 31. Elementi contenuti nell'elemento <deactivateOnEvent>

Elemento	Obbligatorio o facoltativo?
<eventType>	Facoltativo. Sono consentite zero o più ricorrenze.
<filteringPredicate>	Facoltativo. E' consentito 0 o 1.

Elemento duplicateRule

L'elemento <duplicateRule> definisce una regola in base allo schema di duplicato.

Attributi

<duplicateRule> contiene i seguenti attributi:

Tabella 32. Attributi dell'elemento <duplicateRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No

Contenuto in

<duplicateRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<duplicateRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 33. Elementi contenuti nell'elemento <duplicateRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<eventSelector>	Facoltativo. E' consentito 0 o 1.
<groupingKey>	Facoltativo. E' consentito 0 o 1.
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onDetection>	Facoltativo. E' consentito 0 o 1.
<onNextEvent>	Facoltativo. E' consentito 0 o 1.
<onTimeWindowComplete>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

“Schema di duplicato” a pagina 12

Una regola di duplicato viene definita da uno schema di duplicato. Esso conta gli eventi secondi e successivi accettati all’interno di un intervallo di tempo specificato ma ignora l’elaborazione della serie di regole per questi eventi. Essa è una regola stateful.

Elemento eventAttribute

L’elemento <eventAttribute> fornisce un modo per associare un tipo di evento ed un attributo di evento come parte del nome alias di attributo che è definito dall’elemento <attributeAlias>.

Attributi

<eventAttribute> ha i seguenti attributi:

Tabella 34. Attributi dell’elemento <eventAttribute>

Nome	Descrizione	Tipo dati	Obbligatorio
type	Definisce il nome del tipo di evento. E’ lo stesso nome utilizzato per l’attributo type sull’elemento <eventType>.	xsd:NMTOKEN	Yes
attributeName	Specifica il nome completo dell’attributo di evento associato agli altri attributi evento tramite il nome alias dell’attributo. Questo nome deve corrispondere al nome utilizzato nella variabile act_event per effettuare una chiamata al metodo getAttribute.	xsd:string	Yes

Contenuto in

<eventAttribute> è contenuto nel seguente elemento:

- <attributeAlias>

Contiene

<eventAttribute> non contiene elementi.

Elemento eventCountThreshold

L’elemento <eventCountThreshold> è valido solo per la regola di soglia. Esso definisce il numero di eventi che devono rispondere ai criteri di selezione eventi in un certo periodo di tempo. L’elemento <eventCountThreshold> specifica anche uno dei seguenti due possibili modi di intervallo di tempo, fisso o scorrevole, per la finestra di tempo.

Dettagli

intervallo fisso

Un intervallo fisso inizia quando viene ricevuto il primo evento che risponde ai criteri di selezione evento e termina quando si verifica una delle condizioni di seguito riportate.

- La regola raggiunge la soglia nella durata di tempo specificata.
- La durata di tempo specificata è trascorsa.

intervallo di scorrimento

Un intervallo scorrevole inizia quando viene ricevuto il primo evento che risponde ai criteri di selezione evento. Tuttavia quando la regola non raggiunge la soglia e la durata specificata è trascorsa, la finestra di tempo regola l'ora di inizio sull'ora di ricezione evento per un nuovo "primo" evento che è in genere l'evento successivo che viene accettato. L'intervallo scorrevole continua a regolarsi in questo modo fino a che non si verifica una delle seguenti condizioni:

- La regola raggiunge la soglia nella durata di tempo specificata.
- Dopo che viene ricevuto l'evento che avvia la finestra di tempo, non vengono ricevuti eventi susseguenti all'interno della durata di tempo specificata.

L'evento che apre la finestra di tempo (diventa il nuovo "primo" evento) è l'evento con un'ora di ricezione che risponde a questi criteri: l'ora di ricezione, aggiunta alla durata dell'intervallo di tempo, è successiva all'ora corrente. Di seguito sono riportati i criteri nel formato di un'equazione:

*ora di ricezione evento + durata di intervallo
di tempo per la regola > ora corrente*

Quando non è presente alcun evento di questo tipo, l'intervallo scorrevole non è in grado di regolare l'ora in alcun modo e l'intervallo termina.

La regola di soglia conteggia ogni evento accettato fino a che non viene raggiunta la soglia o il periodo di tempo termina. Le azioni definite nell'elemento <onDetection> o nell'elemento <onTimeOut> vengono poi eseguite nel modo appropriato.

Azioni <onDetection>

Queste azioni vengono eseguite quando il conteggio eventi è uguale al valore definito dall'attributo threshold dell'elemento <eventCountThreshold> che indica che la soglia è raggiunta.

Azioni <onTimeOut>

Quando queste azioni sono eseguite dipende dal modo di intervallo, se fisso o scorrevole.

modo fisso

In modo fisso le azioni vengono eseguite quando scade la finestra di tempo.

modo scorrevole

In modo scorrevole, queste azioni vengono eseguite se, dopo che viene ricevuto l'evento che avvia la finestra di tempo, non vengono ricevuti eventi susseguenti nella durata di tempo specificata. In altre parole, non vengono ricevuti eventi con un'ora di ricezione che, quando è aggiunta alla durata dell'intervallo di tempo, è successiva all'ora corrente.

Il modo intervallo di tempo per la finestra dell'ora è definita dall'attributo timeIntervalMode dell'elemento <eventCountThreshold>. Lo scenario che segue illustra il comportamento di, e le differenze tra, due possibili modi di intervallo di tempo.

Scenario che illustra i modi fisso e scorrevole

La regola riceve quattro eventi che rispondono ai criteri di selezione eventi, un evento in ognuna di queste ore: 8:00, 8:04, 8:06 e 8:07. La soglia di conteggio eventi è 3 e la durata per la finestra di tempo è 5 minuti.

Comportamento della regola con modo fisso

Con questo modo di intervallo, la regola di soglia inizia l'elaborazione alle 8:00 ed esegue le azioni di <onTimeOut> alle 8:05 in quanto riceve solo 2 eventi in 5 minuti. Di conseguenza, non raggiunge la soglia nella finestra di tempo. Quando alle 8:06 viene ricevuto il terzo evento, la regola di soglia inizia di nuovo l'elaborazione e vengono eseguite le azioni <onTimeOut> alle 8:11 in quanto si ricevono solo 2 eventi in 5 minuti.

Il modo fisso è statico.

Comportamento della regola con modo scorrevole

Con questo modo di intervallo, la regola di soglia inizia l'elaborazione alle 8:00. Alle 8:05, quando è pianificato il completamento della finestra, la regola determina che ha ricevuto solo 2 eventi. La regola elimina poi l'evento che è stato ricevuto alle 8:00 e ricalcola la durata che termina alle 8:09 (in quanto il primo evento è quello ricevuto alle 8:04). Quando la regola riceve l'evento alle 8:07, vengono eseguite le azioni <onDetection> in quanto ha raggiunto la soglia (3 eventi, alle 8:04, 8:06 e 8:07) nell'ultima finestra di tempo (8:04 – 8:09).

Il modo scorrevole è dinamico poiché continua a regolare (a scorrere) l'ora di inizio nel tentativo di raggiungere la soglia nella finestra di tempo.

La regola riceve 4 eventi che rispondono ai criteri di selezione eventi, un evento in ognuna di queste ore: 8:00, 8:04, 8:06 e 8:10. La soglia di conteggio eventi è 3 e la durata per la finestra di tempo è 5 minuti.

Comportamento della regola con modo scorrevole

In questo caso, la regola di soglia inizia l'elaborazione alle 8:00. Alle 8:05, quando è pianificato il completamento della finestra, la regola determina che ha ricevuto solo 2 eventi. La regola elimina poi l'evento che è stato ricevuto alle 8:00 e ricalcola la durata che termina alle 8:09 (in quanto il primo evento è quello ricevuto alle 8:04).

Alle 8:09, quando è pianificato il completamento della finestra, la regola determina che ha ricevuto solo 2 eventi. La regola elimina poi l'evento che è stato ricevuto alle 8:04 e ricalcola la durata che termina alle 8:11 (in quanto il primo evento è quello ricevuto alle 8:06).

Alle 8:11, quando è pianificato il completamento della finestra, la regola determina che ha ricevuto solo 2 eventi. La regola elimina poi l'evento che è stato ricevuto alle 8:06 e ricalcola la durata che termina alle 8:15 (in quanto il primo evento è quello ricevuto alle 8:10).

Alle 8:15, quando è pianificato il completamento della finestra, la regola determina che non ha ricevuto eventi dall'inizio della finestra di tempo alle 8:10. La regola elabora poi le azioni <onTimeOut>.

Attributi

<eventCountThreshold> ha i seguenti attributi:

Tabella 35. Attributi dell'elemento <eventCountThreshold>

Nome	Descrizione	Tipo dati	Obbligatorio
soglia	Definisce il numero di eventi che devono rispondere ai criteri di selezione eventi in un certo periodo di tempo. Definisce il valore di soglia che deve essere raggiunto. Questo valore deve essere un numero intero positivo.	xsd:positiveInteger	Yes
timeIntervalMode	Definisce se l'intervallo di tempo per la finestra di tempo è fisso o scorrevole. I valori validi per questo attributo sono: <ul style="list-style-type: none">• fisso (valore predefinito)• scorrevole	xsd:string	No

Contenuto in

<eventCountThreshold> è contenuto nel seguente elemento:

- <thresholdRule>

Contiene

<eventCountThreshold> non contiene elementi.

Elemento eventSelector

L'elemento <eventSelector> definisce gli eventi selezionati per l'elaborazione da parte di una regola.

Dettagli

Di seguito vengono illustrati tre modi possibili di selezionare gli eventi:

- L'uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L'uso di uno o più elementi <eventType> con un elemento <filteringPredicate>
- L'uso di un elemento <filteringPredicate> senza elementi <eventType>

Nei casi in cui si desidera elaborare tutti gli eventi, sono disponibili le seguenti opzioni:

- Non codificare alcun elemento <eventSelector>.
- Codificare un elemento <eventSelector> che non contiene elementi.

Se non si codifica un elemento <eventType> le prestazioni possono risentirne in modo negativo.

Presumere il caso in cui si selezionano tutti gli eventi del tipo Audit Failure. E' possibile utilizzare un predicato di filtro per restringere i criteri di scelta in modo da includere solo gli eventi che presentano un attributo di evento con un certo valore. Codificare, ad esempio, un elemento <eventType> per selezionare tutti gli

eventi del tipo Audit Failure e codificare un elemento <filteringPredicate> per selezionare solo quegli eventi che hanno un attributo hostname con il valore MyCriticalSystem.

Attributi

<eventSelector> ha il seguente attributo:

Tabella 36. Attributi dell'elemento <eventSelector>

Nome	Descrizione	Tipo dati	Obbligatorio
alias	Questo attributo è valido solo all'interno di una regola di sequenza, che è la sola regola che ha più elementi <eventSelector>. Esso definisce in modo univoco un evento selezionato nella regola di sequenza. I predicati e le azioni di filtro possono quindi utilizzare questi nomi alias per accedere a quell'evento.	xsd:NMTOKEN	No

Contenuto in

<eventSelector> è contenuto nei seguenti elementi:

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

Contiene

<eventSelector> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 37. Elementi contenuti nell'elemento <eventSelector>

Elemento	Obbligatorio o facoltativo?
<eventType>	Facoltativo. Sono consentite zero o più ricorrenze.
<filteringPredicate>	Facoltativo. E' consentito 0 o 1.

Elemento eventType

L'elemento <eventType> definisce il tipo di evento selezionato per l'elaborazione da parte di una regola o l'attivazione/disattivazione della regola.

Attributi

<eventType> ha il seguente attributo:

Tabella 38. Attributi dell'elemento <eventType>

Nome	Descrizione	Tipo dati	Obbligatorio
type	<p>Definisce il tipo di evento.</p> <p>Per gli eventi conformi alla specifica CBE (Common Base Event), questo nome è il valore dell'attributo extensionName.</p> <p>Per gli eventi di IBM Tivoli Enterprise Console, questo nome è il nome di classe di evento definito nel file BAROC.</p> <p>Gli eventi basati su altri formati possono utilizzare un attributo diverso per specificare il tipo di evento.</p>	xsd:NMTOKEN	Yes

Contenuto in

<eventType> è contenuto nei seguenti elementi:

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

Contiene

<eventType> non contiene elementi.

Elemento filteringPredicate

L'elemento <filteringPredicate> contiene un'espressione che limita ulteriormente gli eventi selezionati per l'elaborazione da parte della regola o selezionati per attivare o disattivare la regola. Di conseguenza, gli eventi possono essere filtrati in modo più completo che solo tramite l'elemento <eventType>.

Dettagli

L'espressione definisce una condizione e restituisce un valore booleano, true (la condizione è effettiva) o false (la condizione non è effettiva).

Fare riferimento a "Variabili" a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<filteringPredicate> ha il seguente attributo:

Tabella 39. Attributi dell'elemento <filteringPredicate>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes

Contenuto in

<filteringPredicate> è contenuto nei seguenti elementi:

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

Contiene

<filteringPredicate> non contiene elementi.

Concetti correlati

"Espressioni" a pagina 20

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento filterRule

L'elemento <filterRule> definisce una regola in base allo schema di filtro.

Attributi

<filterRule> contiene i seguenti attributi:

Tabella 40. Attributi dell'elemento <filterRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes

Tabella 40. Attributi dell'elemento <filterRule> (Continua)

Nome	Descrizione	Tipo dati	Obbligatorio
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No

Contenuto in

<filterRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<filterRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 41. Elementi contenuti nell'elemento <filterRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<eventSelector>	Facoltativo. E' consentito 0 o 1.
<onDetection>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema di filtro" a pagina 13

La regola di filtro viene definita dallo schema di filtro. Essa intraprende una certa azione quando accetta un evento. Essa agisce su un singolo evento ed è perciò una regola senza definizione di stato.

Elemento groupingKey

In genere ogni regola attiva ha un'istanza della regola o una copia che è in esecuzione nell'engine dell'Active Correlation Technology. Tuttavia, talvolta la stessa regola è necessaria per gruppi diversi di eventi, spesso correlati ai diversi gruppi di risorse. La chiave di raggruppamento è uno o più attributi di evento o parti di attributi di evento, che possono essere utilizzati per separare gli eventi selezionati in gruppi differenti per l'elaborazione univoca come un gruppo. L'elemento <groupingKey> definisce la chiave di raggruppamento per una regola. Lo scopo di <groupingKey> è di indirizzare una regola in modo da creare un'istanza della regola separata (o una sua copia) per ciascun gruppo di eventi che condivide caratteristiche comuni (come definito dai valori degli attributi che comprendono la chiave di raggruppamento).

Dettagli

I seguenti due scenari illustrano l'importanza della chiave di raggruppamento.

Scenario 1:

Si verificano due eventi, un evento DB2down e un evento DB2up. Il programma DB2 viene eseguito su tre computer chiamati A, B e C. Viene definita una regola di sequenza per correlare un evento DB2down con un evento DB2up e per segnalare all'operatore quando il programma DB2 si interrompe e non viene riavviato.

Se la regola di sequenza viene definita senza la chiave di raggruppamento e viene ricevuto un evento DB2down dal computer A, un evento DB2up da qualsiasi computer completerebbe la sequenza, ma ciò non perseguirebbe lo scopo principale. Tuttavia, se la chiave di raggruppamento è stata definita come attributo hostname, verrà creata una copia univoca, o un'istanza, della regola per ciascun valore univoco dell'attributo hostname negli eventi selezionati. L'engine dell'Active Correlation Technology invia ciascun evento all'istanza della regola corretta (l'istanza della regola per il valore hostname dell'evento). Di conseguenza, se un evento DB2down viene ricevuto dal computer A, l'engine dell'Active Correlation Technology crea un'istanza della regola per il computer A. Se un evento DB2down viene ricevuto dal computer B, l'engine dell'Active Correlation Technology crea una seconda istanza della regola per il computer B. Quando viene ricevuto un evento DB2up dal computer B, l'engine dell'Active Correlation Technology elabora quell'evento nella seconda istanza della regola. La sequenza è completa e l'operatore viene avvisato in quanto gli eventi DB2down e DB2up dal computer B vengono correlati correttamente.

Scenario 2:

Si verifica un evento di messaggio di tentativo di collegamento non corretto su tutti i computer in un ambiente particolare. L'evento contiene un ID utente. Viene definita una regola di soglia per emettere un'avvertenza all'operatore, se questo evento si verifica più di 10 volte in 5 minuti.

Una chiave di raggruppamento può essere definita come ID utente. Poi viene creata una nuova istanza della regola per ogni ID utente univoco. Ogni tentativo di collegamento dell'utente viene registrato in un'istanza della regola univoca di soglia con ogni istanza che ha un conto separato del numero di tentativi di collegamento per quell'utente. L'operatore riceve un'avvertenza se ogni ID utente supera 10 collegamenti non corretti in 5 minuti.

Altre variazioni di questo concetto includono:

- Una chiave di raggruppamento può essere definita come nome host piuttosto che come ID utente. Questa opzione può rilevare un numero esteso di tentativi di collegamento non corretti su un singolo computer.
- La chiave di raggruppamento può essere definita come combinazione di nome host e ID utente. Questa opzione può rilevare un potenziale tentativo di inserimento illegittimo tramite un ID utente specifico su un computer specifico.

Se lo stesso attributo è in tutti i tipi di evento specificati per una regola, l'uso dell'elemento <attributeName> è il modo più semplice e più comune per definire una chiave di raggruppamento.

Attributi

<groupingKey> ha il seguente attributo:

Tabella 42. Attributi dell'elemento <groupingKey>

Nome	Descrizione	Tipo dati	Obbligatorio
missingAttributeHandling	<p>Definisce l'azione che la regola deve intraprendere in ognuna di queste condizioni:</p> <ul style="list-style-type: none">• Quando un evento selezionato ha un attributo che fa parte della chiave di raggruppamento ma il valore per quell'attributo manca.• Quando l'espressione nell'elemento <computedValue> restituisce un valore nullo. La regola tratta questo valore nullo come un valore di attributo mancante. <p>I valori validi per l'attributo missingAttributeHandling sono:</p> <ul style="list-style-type: none">• ignoreEvent (valore predefinito), che significa che la regola ignora l'evento e non intraprende alcuna azione.• ignoreAttribute, che indica che la regola accetta l'evento ma ignora l'attributo con il valore mancante. l'engine dell'Active Correlation Technology include poi un valore sostituto per l'attributo.	xsd:string	No

Contenuto in

<groupingKey> è contenuto nei seguenti elementi:

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

Contiene

<groupingKey> contiene i seguenti elementi.

Tabella 43. Elementi contenuti nell'elemento <groupingKey>

Elemento	Obbligatorio o facoltativo?
<attributeAlias>	E' richiesto 1 di questi elementi. E' facoltativo codificare più di 1 di questi elementi. Sono consentite più ricorrenze di tutti e tre gli elementi. Questi elementi possono essere codificati in un ordine qualsiasi.
<attributeName>	
<computedValue>	

Elemento import

L'elemento <import> contiene un'espressione che specifica i moduli esterni (come le classi Java) da importare per il loro utilizzo in altre espressioni all'interno delle regole.

Dettagli

Il codice di espressione è una stringa all'interno dell'elemento <import>. Il compilatore Active Correlation Technology utilizza le istruzioni import fornite dagli elementi <import> per compilare il codice di espressione all'interno delle regole che effettuano chiamate ai metodi esterni.

Attributi

<import> ha il seguente attributo:

Tabella 44. Attributi dell'elemento <import>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes

Contenuto in

<import> è contenuto nei seguenti elementi:

- <ruleSet>
- <ruleBlock>

Contiene

<import> non contiene elementi.

Concetti correlati

"Importazione e accesso ai moduli e oggetti esterni" a pagina 21

Questo esempio indica come è possibile rendere il codice esterno (come le classi Java) e gli oggetti esterni accessibili alle espressioni. Un oggetto esterno è un oggetto che un'applicazione crea per comunicare con le espressioni.

Elemento inactiveWhenLoaded

L'elemento <inactiveWhenLoaded> specifica che una regola non è attiva quando viene caricata dall'engine Active Correlation Technology. La regola rimane inattiva fino a che non viene attivata con un altro mezzo.

Attributi

<inactiveWhenLoaded> non ha attributi.

Contenuto in

<inactiveWhenLoaded> è contenuto nel seguente elemento:

- <start>

Contiene

<inactiveWhenLoaded> non contiene elementi.

Elemento lifeCycleActions

L'elemento <lifeCycleActions> contiene gli elementi che definiscono le azioni da intraprendere nelle quattro fasi principali nel ciclo di vita di una regola.

Dettagli

Le azioni che vengono definite per le fasi di caricamento e attivazione vengono richiamate dopo che la regola viene caricata o attivata realmente ma prima che la regola inizi l'elaborazione. Le azioni definite per le fasi di disattivazione e caricamento vengono richiamate immediatamente prima che la regola venga disattivata o scaricata.

Attributi

<lifeCycleActions> non ha attributi.

Contenuto in

<lifeCycleActions> è contenuto nei seguenti elementi:

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

Contiene

<lifeCycleActions> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 45. Elementi contenuti nell'elemento `<lifeCycleActions>`

Elemento	Obbligatorio o facoltativo?
<code><onLoad></code>	Facoltativo. E' consentito 0 o 1.
<code><onActivation></code>	Facoltativo. E' consentito 0 o 1.
<code><onDeactivation></code>	Facoltativo. E' consentito 0 o 1.
<code><onUnload></code>	Facoltativo. E' consentito 0 o 1.

elemento never

L'elemento `<never>` specifica che una regola non viene mai disattivata in un determinato periodo di tempo. Una regola può tuttavia essere disattivata da un evento o da altri mezzi.

Attributi

`<never>` non ha attributi.

Contenuto in

`<never>` è contenuto nel seguente elemento:

- `<stop>`

Contiene

`<never>` non contiene elementi.

Elemento onActivation

L'elemento `<onActivation>` specifica l'azione o la serie di azioni da intraprendere quando la regola viene attivata. L'azione `<onActivation>` viene chiamata dopo che la regola è stata attivata ma prima che la regola inizi l'elaborazione.

Dettagli

Se la serie di regole contiene più regole che vengono attivate alla stessa data e ora o per evento uguale, le azioni di seguito riportate non vengono eseguite alla stessa ora:

- Azioni di risposta alla regola negli elementi `<onTimeOut>` e `<onTimeWindowComplete>`
- Azioni di durata negli elementi `<onActivation>` e `<onDeactivation>`

Queste azioni vengono eseguite in sequenza in qualsiasi ordine. Non vengono necessariamente eseguite nell'ordine in cui vengono codificate nella serie di regole. Poiché ogni azione deve essere completata prima che inizi l'azione successiva nella sequenza, le azioni non vengono eseguite contemporaneamente.

Attributi

`<onActivation>` non ha attributi.

Contenuto in

<onActivation> è contenuto nel seguente elemento:

- <lifeCycleActions>

Contiene

<onActivation> contiene il seguente elemento:

Tabella 46. Elementi contenuti nell'elemento <onActivation>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento onDeactivation

L'elemento <onDeactivation> specifica l'azione o la serie di azioni da intraprendere quando la regola viene disattivata. Viene effettuata una chiamata sull'azione <onDeactivation> prima che la regola viene disattivata.

Dettagli

Se la serie di regole contiene più regole che vengono attivate alla stessa data e ora o per evento uguale, le azioni di seguito riportate non vengono eseguite alla stessa ora:

- Azioni di risposta alla regola negli elementi <onTimeOut> e <onTimeWindowComplete>
- Azioni di durata negli elementi <onActivation> e <onDeactivation>

Queste azioni vengono eseguite in sequenza in qualsiasi ordine. Non vengono necessariamente eseguite nell'ordine in cui vengono codificate nella serie di regole. Poiché ogni azione deve essere completata prima che inizi l'azione successiva nella sequenza, le azioni non vengono eseguite contemporaneamente.

Attributi

<onDeactivation> non ha attributi.

Contenuto in

<onDeactivation> è contenuto nel seguente elemento:

- <lifeCycleActions>

Contiene

<onDeactivation> contiene il seguente elemento:

Tabella 47. Elementi contenuti nell'elemento <onDeactivation>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento onDetection

L'elemento <onDetection> è valido solo per le regole di duplicato, filtro, sequenza e soglia. Esso specifica l'azione o la serie di azioni da intraprendere quando viene rilevato lo schema di regola.

Dettagli

Tabella 48 descrive come viene rilevato lo schema di regola per ogni tipo di regola in cui è valida un'azione <onDetection>.

Tabella 48. Come viene rilevato uno schema di regola in base al tipo di regola

Tipo di regola	Come viene rilevato lo schema di regola
duplicato	Questo schema di regola viene rilevato quando viene ricevuto il primo evento che risponde ai criteri di selezione evento.
filter	Questo schema di regola viene rilevato quando viene ricevuto un evento che risponde ai criteri di selezione evento.
sequenza	Questo schema di regola viene rilevato quando viene ricevuta una sequenza di eventi che risponde ai criteri di selezione evento nell'ordine appropriato e in una finestra di tempo.
soglia	Questo schema di regola viene rilevato quando viene ricevuta una sequenza di eventi che risponde ai criteri di selezione evento in una finestra di tempo e al valore di soglia.

Attributi

<onDetection> non ha attributi.

Contenuto in

<onDetection> è contenuto nei seguenti elementi:

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

Contiene

<onDetection> contiene il seguente elemento:

Tabella 49. Elementi contenuti nell'elemento <onDetection>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento onLoad

L'elemento <onLoad> specifica l'azione o la serie di azioni da intraprendere quando la regola viene caricata (o distribuita) nell'engine dell'Active Correlation Technology in esecuzione. L'azione <onLoad> viene chiamata dopo che la regola viene caricata ma prima di iniziare qualsiasi elaborazione.

Attributi

<onLoad> non ha attributi.

Contenuto in

<onLoad> è contenuto nel seguente elemento:

- <lifeCycleActions>

Contiene

<onLoad> contiene il seguente elemento:

Tabella 50. Elementi contenuti nell'elemento <onLoad>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento onNextEvent

L'elemento <onNextEvent> è valido solo per la regola di duplicato. Esso specifica l'azione o la serie di azioni da intraprendere quando la regola di duplicato riceve il secondo ed ogni successivo evento che risponde ai criteri di selezione evento, nella finestra di tempo specificata.

Dettagli

Per le regole di duplicato, l'engine dell'Active Correlation Technology ignora l'elaborazione della serie di regole per il secondo e per ogni successivo evento che risponde ai criteri di selezione evento nella finestra di tempo specificata. Di conseguenza l'unico motivo per la codifica di un'azione <onNextEvent> è quello di specificare l'elaborazione alternativa per il secondo ed ogni evento successivo.

Attributi

<onNextEvent> non ha attributi.

Contenuto in

<onNextEvent> è contenuto nel seguente elemento:

- <duplicateRule>

Contiene

<onNextEvent> contiene il seguente elemento:

Tabella 51. Elementi contenuti nell'elemento <onNextEvent>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento onTimeOut

L'elemento <onTimeOut> è valido solo per le regole di sequenza e di soglia. Esso specifica l'azione o la serie di azioni da intraprendere quando scade la finestra di tempo per la regola.

Dettagli

La Tabella 52 descrive come scade la finestra di tempo per ogni tipo di regola in cui è valida un'azione <onTimeOut>.

Tabella 52. Scadenza di una finestra di tempo in base al tipo di regola

Tipo di regola	Scadenza di una finestra di tempo
sequenza	La finestra di tempo scade se uno o più eventi vengono accettati ma la sequenza completa degli eventi non viene ricevuta nella finestra di tempo.
soglia	La finestra di tempo scade se uno o più eventi vengono accettati ma la soglia non viene rilevata nella finestra di tempo.

Se la serie di regole contiene più regole che vengono attivate alla stessa data e ora o per evento uguale, le azioni di seguito riportate non vengono eseguite alla stessa ora:

- Azioni di risposta alla regola negli elementi <onTimeOut> e <onTimeWindowComplete>
- Azioni di durata negli elementi <onActivation> e <onDeactivation>

Queste azioni vengono eseguite in sequenza in qualsiasi ordine. Non vengono necessariamente eseguite nell'ordine in cui vengono codificate nella serie di regole. Poiché ogni azione deve essere completata prima che inizi l'azione successiva nella sequenza, le azioni non vengono eseguite contemporaneamente.

Attributi

<onTimeOut> non ha attributi.

Contenuto in

<onTimeOut> è contenuto nei seguenti elementi:

- <sequenceRule>
- <thresholdRule>

Contiene

<onTimeOut> contiene il seguente elemento:

Tabella 53. Elementi contenuti nell'elemento <onTimeOut>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento `onTimeWindowComplete`

L'elemento `<onTimeWindowComplete>` è valido solo per le regole di raccolta, di calcolo, di duplicato di timer. Esso specifica l'azione o la serie di azioni da intraprendere quando è chiusa la finestra di tempo per la regola.

Dettagli

Se la serie di regole contiene più regole che vengono attivate alla stessa data e ora o per evento uguale, le azioni di seguito riportate non vengono eseguite alla stessa ora:

- Azioni di risposta alla regola negli elementi `<onTimeOut>` e `<onTimeWindowComplete>`
- Azioni di durata negli elementi `<onActivation>` e `<onDeactivation>`

Queste azioni vengono eseguite in sequenza in qualsiasi ordine. Non vengono necessariamente eseguite nell'ordine in cui vengono codificate nella serie di regole. Poiché ogni azione deve essere completata prima che inizi l'azione successiva nella sequenza, le azioni non vengono eseguite contemporaneamente.

Attributi

`<onTimeWindowComplete>` non ha attributi.

Contenuto in

`<onTimeWindowComplete>` è contenuto nei seguenti elementi:

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

Contiene

`<onTimeWindowComplete>` contiene il seguente elemento:

Tabella 54. Elementi contenuti nell'elemento `<onTimeWindowComplete>`

Elemento	Obbligatorio o facoltativo?
<code><action></code>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento `onUnload`

L'elemento `<onUnload>` specifica l'azione o la serie di azioni da intraprendere quando la regola viene scaricata o rimossa dall'engine Active Correlation Technology in esecuzione. Sull'azione `<onUnload>` viene effettuata una chiamata prima che la regola viene scaricata.

Attributi

`<onUnload>` non ha attributi.

Contenuto in

<onUnload> è contenuto nel seguente elemento:

- <lifeCycleActions>

Contiene

<onUnload> contiene il seguente elemento:

Tabella 55. Elementi contenuti nell'elemento <onUnload>

Elemento	Obbligatorio o facoltativo?
<action>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento ruleBlock

L'elemento <ruleBlock> fornisce il modo di raggruppare le regole correlate e di organizzare le regole in una gerarchia.

Attributi

<ruleBlock> ha il seguente attributo:

Tabella 56. Attributi dell'elemento <ruleBlock>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica il blocco di regole. Questo identificativo deve essere univoco all'interno della serie di regole o del blocco di regole che contiene questo blocco di regole. Non può contenere un punto.	xsd:NMTOKEN	Yes

Contenuto in

<ruleBlock> è contenuto nei seguenti elementi:

- <ruleSet>
- <ruleBlock>

Contiene

<ruleBlock> contiene i seguenti elementi.

Se essi sono codificati, gli elementi <comment>, <import> e <variable> devono essere codificati nell'ordine riportato. Gli elementi rimanenti possono essere codificati in un ordine qualsiasi.

Tabella 57. Elementi contenuti nell'elemento <ruleBlock>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<import>	Facoltativo. Sono consentite zero o più ricorrenze.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<ruleBlock>	Facoltativo. Sono consentite zero o più ricorrenze.

Tabella 57. Elementi contenuti nell'elemento <ruleBlock> (Continua)

Elemento	Obbligatorio o facoltativo?
<collectionRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<computationRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<duplicateRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<filterRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<sequenceRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<thresholdRule>	Facoltativo. Sono consentite zero o più ricorrenze.
<timerRule>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento ruleSet

L'elemento <ruleSet> definito tramite `act:ruleSet` è l'elemento principale per il linguaggio delle regole dell'Active Correlation Technology. Tutti gli altri elementi sono contenuti all'interno di questo elemento <ruleSet>.

Dettagli

Gli elementi <ruleSet> definiti dallo schema del linguaggio dell'Active Correlation Technology (`act:ruleSet`) e dallo schema di serie di regole di base dell'Active Correlation Technology (`br:ruleSet`) sono duplicati. Tuttavia, quando si crea una serie di regole, è necessario specificare il seguente spazio per i nomi sull'elemento <ruleSet>: `act:ruleSet`.

Attributi

<ruleSet> ha il seguente attributo:

Tabella 58. Attributi dell'elemento <ruleSet>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la serie di regole. Questo identificatore deve essere univoco. Non può contenere un punto.	xsd:NMTOKEN	Yes

Contenuto in

Poiché <ruleSet> è l'elemento principale per il linguaggio di regole, esso non è contenuto all'interno di qualsiasi elemento.

Contiene

<ruleSet> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 59. Elementi contenuti nell'elemento <ruleSet>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.

Tabella 59. Elementi contenuti nell'elemento <ruleSet> (Continua)

Elemento	Obbligatorio o facoltativo?
<import>	Facoltativo. Sono consentite zero o più ricorrenze.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<ruleBlock>	Facoltativo. Sono consentite zero o più ricorrenze.

Elemento runUntilDeactivated

L'elemento <runUntilDeactivated> specifica che la finestra di tempo continua ad essere aperta fino a che la regola non viene disattivata. Di conseguenza, la finestra di tempo per questa regola inizia quando la regola avvia l'elaborazione e non si blocca fino a che la regola non viene disattivata o rimossa dalla serie di regole o se l'engine dell'Active Correlation Technology viene chiuso.

Dettagli

Il comportamento specifico di una regola che include l'elemento <runUntilDeactivated> dipende dal tipo di regola. La Tabella 60 descrive il comportamento della regola per ogni tipo di regola in cui è valido l'elemento <timeWindow> e contiene l'elemento <runUntilDeactivated>.

Tabella 60. Comportamento della regola quando viene attivato <runUntilDeactivated>

Tipo di regola	Comportamento della regola quando viene attivato <runUntilDeactivated>
raccolta	La regola di raccolta accetta il primo evento che risponde ai criteri di selezione evento e continua ad accettare ed elaborare gli eventi fino a che la regola non viene disattivata, nel momento in cui vengono eseguite le azioni definite all'interno dell'elemento <onTimeWindowComplete>, seguite immediatamente dalle azioni che sono definite nell'elemento <onDeactivation>.
calcolo	La regola di calcolo accetta il primo evento che risponde ai criteri di selezione evento e continua ad accettare ed elaborare gli eventi fino a che la regola non viene disattivata, nel momento in cui vengono eseguite le azioni definite all'interno dell'elemento <onTimeWindowComplete>, seguite immediatamente dalle azioni che sono definite nell'elemento <onDeactivation>.
duplicato	La regola di duplicato accetta il primo evento che risponde ai criteri di selezione evento e continua ad accettare ed elaborare gli eventi fino a che la regola non viene disattivata, nel momento in cui vengono eseguite le azioni definite all'interno dell'elemento <onTimeWindowComplete>, seguite immediatamente dalle azioni che sono definite nell'elemento <onDeactivation>.
sequenza	<p>La regola di sequenza accetta il primo evento che risponde ai criteri di selezione evento e continua ad accettare ed elaborare gli eventi fino a che non si verifica una delle seguenti circostanze:</p> <ul style="list-style-type: none"> Viene rilevato lo schema di sequenza. Quando ciò si verifica, le azioni definite nell'elemento <onDetection> vengono eseguite e la regola ritorna allo stato iniziale. L'elaborazione di evento di questa regola inizia di nuovo e questo processo può essere ripetuto molte volte fino a che la regola non viene disattivata. La regola viene disattivata mentre sta elaborando gli eventi. Quando ciò si verifica, le azioni definite nell'elemento <onTimeOut> vengono eseguite, immediatamente seguite dalle azioni definite nell'elemento <onDeactivation>.

Tabella 60. Comportamento della regola quando viene attivato
<runUntilDeactivated> (Continua)

Tipo di regola	Comportamento della regola quando viene attivato <runUntilDeactivated>
soglia	<p>La regola di soglia accetta il primo evento che risponde ai criteri di selezione evento e continua ad accettare ed elaborare gli eventi fino a che non si verifica una delle seguenti circostanze:</p> <ul style="list-style-type: none"> • Viene rilevato lo schema di soglia. Quando ciò si verifica, le azioni definite nell'elemento <onDetection> vengono eseguite e la regola ritorna allo stato iniziale. L'elaborazione di evento di questa regola inizia di nuovo e questo processo può essere ripetuto molte volte fino a che la regola non viene disattivata. • La regola viene disattivata mentre sta elaborando gli eventi. Quando ciò si verifica, le azioni definite nell'elemento <onTimeOut> vengono eseguite, immediatamente seguite dalle azioni definite nell'elemento <onDeactivation>.
timer	Dopo che la regola timer diventa attiva, non fa nulla fino a che non viene disattivata nel momento in cui vengono eseguite le azioni definite all'interno dell'elemento <onTimeWindowComplete>, seguite immediatamente dalle azioni che sono definite nell'elemento <onDeactivation>. L'attributo repeat sull'elemento <timerRule> viene ignorato.

Attributi

<runUntilDeactivated> non ha attributi.

Contenuto in

<runUntilDeactivated> è contenuto nel seguente elemento:

- <timeWindow>

Contiene

<runUntilDeactivated> non contiene elementi.

Elemento sequenceRule

L'elemento <sequenceRule> definisce una regola in base allo schema di sequenza. La regola della sequenza è l'unica regola che consente più programmi di selezione evento. Essa richiede anche un minimo di due programmi di selezione evento.

Attributi

<sequenceRule> ha i seguenti attributi:

Tabella 61. Attributi dell'elemento <sequenceRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes

Tabella 61. Attributi dell'elemento <sequenceRule> (Continua)

Nome	Descrizione	Tipo dati	Obbligatorio
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No
arrivalOrder	Definisce se gli eventi devono arrivare nell'ordine in cui gli elementi <eventSelector> vengono codificati per la regola. I valori validi sono: <ul style="list-style-type: none"> • inOrder (il valore predefinito) • randomOrder 	xsd:string	No

Se il valore dell'attributo arrivalOrder è randomOrder, l'ordine in cui vengono codificati gli elementi <eventSelector> è importante. Gli elementi <eventSelector> con criteri di selezione evento più specifici devono essere codificati prima degli elementi <eventSelector> con criteri di selezione evento meno specifici. In caso contrario la sequenza non viene rilevata quando dovrebbe.

Ad esempio:

- Vengono definiti tre elementi <eventSelector>.
- Il primo elemento <eventSelector> ricerca l'evento eventA.
- Il secondo elemento <eventSelector> ricerca un evento qualsiasi.
- Il terzo elemento <eventSelector> ricerca l'evento eventB.
- I seguenti eventi vengono presentati al sistema nella finestra di tempo specificata: eventA, eventB, eventC.

Il comportamento della regola è il seguente e la sequenza non viene rilevata quando dovrebbe:

1. Il primo evento, eventA, viene accettato dal primo elemento <eventSelector>.
2. Il secondo evento, eventB, viene accettato dal secondo elemento <eventSelector>.
3. Il terzo evento, eventC, viene ignorato.

Lo scenario che segue prevede che gli elementi <eventSelector> sono stati codificati correttamente, con i criteri di selezione evento più specifici che precedono i criteri di selezione evento meno specifici:

- Il primo elemento <eventSelector> ricerca l'evento eventA.
- Il secondo elemento <eventSelector> ricerca l'evento eventB.
- Il terzo elemento <eventSelector> ricerca un evento qualsiasi.

Il comportamento della regola è il seguente e la sequenza viene rilevata:

1. Il primo evento, eventA, viene accettato dal primo elemento <eventSelector>.
2. Il secondo evento, eventB, viene accettato dal secondo elemento <eventSelector>.
3. Il terzo evento, eventC, viene accettato dal terzo elemento <eventSelector>.

Contenuto in

<sequenceRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<sequenceRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 62. Elementi contenuti nell'elemento <sequenceRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<eventSelector>	Sono richieste 2 ricorrenze di questo elemento per la regola di sequenza. Sono consentite ulteriori ricorrenze.
<groupingKey>	Facoltativo. E' consentito 0 o 1.
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onDetection>	Facoltativo. E' consentito 0 o 1.
<onTimeOut>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema di sequenza" a pagina 14

Una regola di sequenza viene definita da uno schema di sequenza. Essa rileva se una certa sequenza di eventi arriva all'interno di un intervallo di tempo. La sequenza può essere ordinata o casuale. Essa è una regola stateful.

Elemento start

L'elemento <start> definisce se una regola viene attivata in un momento specifico o quando viene caricata dall'engine Active Correlation Technology.

Dettagli

Se l'elemento <start> non viene per niente codificato, l'ora di avvio predefinita è la stessa di quella definita dall'elemento <whenLoaded>.

Attributi

<start> non ha attributi.

Contenuto in

<start> è contenuto nel seguente elemento:

- <activationTime>

Contiene

<start> contiene i seguenti elementi:

Tabella 63. Elementi contenuti nell'elemento <start>

Elemento	Obbligatorio o facoltativo?
<dateTime>	E' richiesto uno dei seguenti elementi ed è consentita una sola ricorrenza dell'elemento scelto.
<whenLoaded>	
<inactiveWhenLoaded>	

Elemento stop

L'elemento <stop> definisce se una regola viene disattivata in un momento specifico, dopo una certa durata o mai.

Dettagli

Se l'elemento <stop> non viene per niente codificato, l'ora di arresto predefinita è la stessa di quella definita dall'elemento <never>.

Attributi

L'elemento <stop> non ha attributi.

Contenuto in

L'elemento <stop> è contenuto nel seguente elemento:

- <activationTime>

Contiene

L'elemento <stop> contiene i seguenti elementi:

Tabella 64. Elementi contenuti nell'elemento <stop>

Elemento	Obbligatorio o facoltativo?
<dateTime>	E' richiesto uno dei seguenti elementi ed è consentita una sola ricorrenza dell'elemento scelto.
<never>	
<after>	

Elemento stopAfter

L'elemento <stopAfter> specifica la durata del tempo in cui un'istanza della regola, come definita dall'elemento <groupingKey> deve rimanere attiva dopo la sua attivazione. Dopo questo periodo, l'istanza della regola deve essere disattivata.

Attributi

L'elemento <stopAfter> contiene i seguenti attributi:

Tabella 65. Attributi dell'elemento <stopAfter>

Nome	Descrizione	Tipo dati	Obbligatorio
durata	Specifica la quantità di tempo per la durata. Il tipo di dati di questo attributo dipende dal valore dell'attributo unit.	<ul style="list-style-type: none">Se il valore dell'attributo unit è ISO-8601, il tipo di dati è xsd:duration.Se il valore dell'attributo unit è milliseconds, il tipo di dati è xsd:positiveInteger.	Yes
unit	Specifica l'unità di tempo da utilizzare. I valori validi per questo attributo sono: <ul style="list-style-type: none">ISO-8601milliseconds	xsd:string	Yes

Uso dello standard ISO 8601 per la durata di tempo

La codifica ISO-8601 come valore per l'attributo unit indica che il valore dell'attributo di durata è codificato in base allo standard ISO 8601 per specificare una durata di tempo come una stringa. La specifica del tipo di dati dello schema standard XML utilizza ISO 8601 per fornire un tipo di dati chiamato duration. Questo tipo di dati viene descritto in dettaglio su <http://www.w3.org/TR/xmlschema-2/#duration>.

Il formato per il tipo di dati duration nello schema standard XML è la seguente stringa:

$PnYnMnDnTnHnMnS$

- P è il carattere che dà sempre inizio alla stringa.
- nY rappresenta il numero di anni. Un anno è lo stesso di 365 giorni. Di conseguenza la codifica 1Y è uguale alla codifica 365D.
- nM rappresenta il numero di mesi. Un mese è lo stesso di 30 giorni. Di conseguenza la codifica 1M è uguale alla codifica 30D.
- nD rappresenta il numero di giorni.
- T è il separatore che separa le unità di giorni (anni, mesi e giorni) dalle unità di tempo (ore, minuti e secondi). Le unità seguono sempre T.
- nH rappresenta il numero di ore.
- nM rappresenta il numero di minuti.
- nS rappresenta il numero di secondi.

Di seguito sono riportati esempi del formato:

- P5DT12H è 5,5 giorni.
- PT59M59S è 59 minuti e 59 secondi.
- P1M è 1 mese.

Contenuto in

<stopAfter> è contenuto nell'elemento <activateOnEvent> ma solo quando <activateOnEvent> viene codificato nell'elemento <activationByGroupingKey>.

Contiene

<stopAfter> non contiene elementi.

Elemento thresholdRule

L'elemento <thresholdRule> definisce una regola in base allo schema di soglia.

Attributi

<thresholdRule> contiene i seguenti attributi:

Tabella 66. Attributi dell'elemento <thresholdRule>

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	xsd:NMTOKEN	Yes
processOnlyForwardedEvents	Determina se la regola riceve tutti gli eventi o solo gli eventi inoltrati da altre regole. Il valore predefinito è false, che indica che la regola riceve tutti gli eventi, compreso quelli inoltrati da altre regole.	xsd:boolean	No

Contenuto in

<thresholdRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<thresholdRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 67. Elementi contenuti nell'elemento <thresholdRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.

Tabella 67. Elementi contenuti nell'elemento <thresholdRule> (Continua)

Elemento	Obbligatorio o facoltativo?
<eventSelector>	Facoltativo. E' consentito 0 o 1.
<groupingKey>	Facoltativo. E' consentito 0 o 1.
<booleanThreshold>	E' richiesto uno dei seguenti elementi ed è consentita una sola ricorrenza dell'elemento scelto.
<computedThreshold>	
<eventCountThreshold>	
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onDetection>	Facoltativo. E' consentito 0 o 1.
<onTimeOut>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema di soglia" a pagina 16

La regola di soglia viene definita dallo schema di soglia. Essa raccoglie un gruppo di eventi selezionati in un intervallo di tempo e determina, dopo che è stato ricevuto ogni evento, se viene soddisfatta una condizione di soglia. Essa è una regola stateful.

Elemento timeInterval

L'elemento <timeInterval> specifica la durata della finestra di tempo.

Attributi

<timeInterval> contiene i seguenti attributi:

Tabella 68. Attributi dell'elemento <timeInterval>

Nome	Descrizione	Tipo dati	Obbligatorio
durata	Specifica la quantità di tempo per la durata. Il tipo di dati di questo attributo dipende dal valore dell'attributo unit.	<ul style="list-style-type: none"> Se il valore dell'attributo unit è ISO-8601, il tipo di dati è xsd:duration. Se il valore dell'attributo unit è milliseconds, il tipo di dati è xsd:positiveInteger. 	Yes
unit	Specifica l'unità di tempo da utilizzare. I valori validi per questo attributo sono: <ul style="list-style-type: none"> ISO-8601 milliseconds 	xsd:string	Yes

Uso dello standard ISO 8601 per la durata di tempo

La codifica ISO-8601 come valore per l'attributo unit indica che il valore dell'attributo di durata è codificato in base allo standard ISO 8601 per specificare una durata di tempo come una stringa. La specifica del tipo di dati dello schema standard XML utilizza ISO 8601 per fornire un tipo di dati chiamato duration. Questo tipo di dati viene descritto in dettaglio su <http://www.w3.org/TR/xmlschema-2/#duration>.

Il formato per il tipo di dati `duration` nello schema standard XML è la seguente stringa:

`PnYnMnDTnHnMnS`

- `P` è il carattere che dà sempre inizio alla stringa.
- `nY` rappresenta il numero di anni. Un anno è lo stesso di 365 giorni. Di conseguenza la codifica `1Y` è uguale alla codifica `365D`.
- `nM` rappresenta il numero di mesi. Un mese è lo stesso di 30 giorni. Di conseguenza la codifica `1M` è uguale alla codifica `30D`.
- `nD` rappresenta il numero di giorni.
- `T` è il separatore che separa le unità di giorni (anni, mesi e giorni) dalle unità di tempo (ore, minuti e secondi). Le unità seguono sempre `T`.
- `nH` rappresenta il numero di ore.
- `nM` rappresenta il numero di minuti.
- `nS` rappresenta il numero di secondi.

Di seguito sono riportati esempi del formato:

- `P5DT12H` è 5,5 giorni.
- `PT59M59S` è 59 minuti e 59 secondi.
- `P1M` è 1 mese.

Contenuto in

`<timeInterval>` è contenuto nel seguente elemento:

- `<timeWindow>`

Contiene

`<timeInterval>` non contiene elementi.

Elemento `timerRule`

L'elemento `<timerRule>` definisce una regola in base allo schema di timer.

Attributi

`<timerRule>` contiene i seguenti attributi:

Tabella 69. Attributi dell'elemento `<timerRule>`

Nome	Descrizione	Tipo dati	Obbligatorio
<code>name</code>	Identifica la regola. Questo identificativo deve essere univoco all'interno del blocco di regole che contiene questa regola. Non può contenere un punto.	<code>xsd:NMTOKEN</code>	Yes
<code>processOnlyForwardedEvents</code>	Questo attributo viene ignorato poiché la regola del timer non elabora gli eventi.	<code>xsd:boolean</code>	No

Tabella 69. Attributi dell'elemento <timerRule> (Continua)

Nome	Descrizione	Tipo dati	Obbligatorio
repeat	<p>Definisce se la regola timer viene eseguita in modo ripetuto fino a che non viene disattivata. I valori validi sono:</p> <ul style="list-style-type: none"> • true (il valore predefinito) • false <p>Se il valore è impostato su false, la regola viene eseguita nell'intervallo di tempo solo una volta, esegue l'azione di risposta della regola quando la relativa finestra di tempo è completa e si arresta.</p> <p>Se l'elemento <timeWindow> per la regola timer contiene l'elemento <runUntilDeactivated>, l'attributo repeat viene ignorato.</p>	xsd:boolean	No

Contenuto in

<timerRule> è contenuto nel seguente elemento:

- <ruleBlock>

Contiene

<timerRule> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 70. Elementi contenuti nell'elemento <timerRule>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<variable>	Facoltativo. Sono consentite zero o più ricorrenze.
<activationInterval>	Facoltativo. E' consentito 0 o 1.
<lifeCycleActions>	Facoltativo. E' consentito 0 o 1.
<timeWindow>	Obbligatorio. E' consentita solo una ricorrenza.
<onTimeWindowComplete>	Facoltativo. E' consentito 0 o 1.

Concetti correlati

"Schema timer" a pagina 18

La regola timer viene definita dallo schema timer. Essa avvia le azioni ad intervalli regolari. Essa è una regola stateful. Sebbene una regola timer non elabora gli eventi, essa può essere attivata o disattivata da un evento.

Elemento timeWindow

L'elemento <timeWindow> contiene elementi che definiscono l'intervallo di tempo durante il quale la regola è in elaborazione.

Dettagli

Ad esempio, la finestra di tempo per una regola di duplicato definisce per quanto tempo la regola deve ricercare gli eventi che sono duplicati del primo evento che viene accettato. Se la finestra di tempo è 30 secondi, la regola di duplicato elabora tutti gli eventi duplicati che si verificano nei 30 secondi del primo evento accettato.

Attributi

<timeWindow> non ha attributi.

Contenuto in

<timeWindow> è contenuto nei seguenti elementi:

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

Contiene

<timeWindow> contiene i seguenti elementi:

Tabella 71. Elementi contenuti nell'elemento <timeWindow>

Elemento	Obbligatorio o facoltativo?
<timeInterval>	E' richiesto uno dei seguenti elementi ed è consentita una sola ricorrenza dell'elemento scelto.
<runUntilDeactivated>	

Elemento variable

L'elemento <variable> definisce una variabile e contiene le informazioni in un formato che può essere indicato dalle espressioni. Una variabile può essere definita al livello della serie di regole, di un blocco di regole o di una regola.

Dettagli

Variabile della serie di regole

Si applica globalmente alla serie di regole e può essere indicato da qualsiasi espressione in quella serie di regole.

Variabile del blocco di regole

Si applica solo all'interno del blocco di regole (e all'interno di tutti i blocchi di regole contenuti) e può essere indicata in qualsiasi espressione all'interno del blocco di regole.

Variabile della regola

Si applica solo alle espressioni all'interno di quella regola.

Le variabili possono avere lo stesso nome su livelli diversi nella gerarchia delle regole. Quando si accede a una variabile, viene utilizzata la definizione più locale della variabile. Ad esempio, se una variabile viene definita a livello di serie di regole, di blocco di regole e di regola, la definizione della variabile a livello di regola viene utilizzata dalle espressioni all'interno di quella regola.

Quando le variabili vengono definite al livello della serie di regole o di blocco di regole, più regole prendono e impostano queste variabili in momenti diversi. Di conseguenza, per assicurarsi che i valori della variabile vengano mantenuti correttamente, è necessario sapere come codificare le interazioni tra le variabili nella serie di regole.

Se la variabile viene definita al livello della serie di regole o di blocco di regole, essa non viene nuovamente inizializzata dopo che lo schema di regola viene adattato.

In ognuna di queste condizioni, utilizzare il blocco sulla richiesta e impostazione delle variabili di serie di regole e di blocco di regole per evitare che i valori della variabile vengano impostati in modo non corretto:

- Se la regola timer ottiene o imposta una variabile durante un'azione `<onTimeout>`
- Se l'applicazione in cui è integrato l'engine dell'Active Correlation Technology è con più sottoprocessi

Se una regola viene definita con una chiave di raggruppamento, le variabili di regola che sono definite dall'elemento `<variable>` non sono valide all'interno delle azioni di durata o all'interno di un elemento `<filteringPredicate>` contenuto in un elemento `<activateOnEvent>` o `<deactivateOnEvent>` che è contenuto in un elemento `<activationInterval>`. Ciò in quanto, in questo caso, le variabili di regola si applicano solo a un'istanza della regola e le istanze della regola non esistono al momento in cui queste espressioni vengono eseguite.

Attributi

`<variable>` contiene i seguenti attributi:

Tabella 72. Attributi dell'elemento `<variable>`

Nome	Descrizione	Tipo dati	Obbligatorio
name	Identifica una variabile speciale. Il riferimento a una variabile avviene tramite il suo nome.	xsd:NMTOKEN	Yes
dataType	Identifica il tipo di informazioni che una variabile contiene. Questo deve essere un tipo dati completo come <code>java.lang.String</code> .	xsd:NMTOKEN	Yes

Limitazioni relative al nome per le variabili

I nomi della variabile hanno alcune limitazioni. Di conseguenza il valore dell'attributo `name` sull'elemento `<variable>` presenta i seguenti limiti:

- Esso può includere solo i seguenti caratteri:
 - Lettere A-Z ASCII maiuscole. La rappresentazione Unicode è `\u0041-\u005a`.
 - Lettere A-Z ASCII minuscole. La rappresentazione Unicode è `\u0061-\u007a`.

- Sottolineatura ASCII (_). La rappresentazione Unicode è \u005f.
- Segno del dollaro (\$). La rappresentazione Unicode è \u0024.
- Cifre ASCII 0 – 9. La rappresentazione Unicode è \u0030-\u0039.
- Non può essere nullo.
- Non può essere una stringa vuota.
- Non può contenere spazi.
- Non può contenere un punto.
- Non può iniziare con act_ in qualsiasi formato (non in maiuscolo, minuscolo o entrambi).

Contenuto in

<variable> è contenuto nei seguenti elementi:

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

Contiene

<variable> contiene i seguenti elementi.

Gli elementi devono essere codificati nell'ordine visualizzato. Se un elemento è facoltativo, non deve essere codificato ma tutti gli elementi codificati devono seguire l'ordine corretto.

Tabella 73. Elementi contenuti nell'elemento <variable>

Elemento	Obbligatorio o facoltativo?
<comment>	Facoltativo. E' consentito 0 o 1.
<varInitializer>	Obbligatorio. E' consentita una ricorrenza.

Concetti correlati

“Variabili” a pagina 24

Nel linguaggio della regola, certe variabili vengono utilizzate per memorizzare le informazioni correlate agli eventi in diverse ricorrenze o regole. A queste informazioni correlate all'evento è possibile accedere dalle espressioni all'interno delle regole. Alcuni tipi di variabili vengono definiti dal programma di scrittura della regola e altri sono forniti dall'Active Correlation Technology. Ad alcuni tipi è possibile accedere direttamente all'interno di un'espressione ed ad altri solo tramite i metodi che sono forniti dall'Active Correlation Technology.

Elemento varInitializer

L'elemento <varInitializer> contiene un'espressione che fornisce il valore iniziale per la variabile definito nell'elemento <variable> associato.

Dettagli

Poiché la variabile può essere di qualsiasi tipo, il codice di espressione può restituire un oggetto array o qualsiasi altro oggetto complesso o specifico dell'implementazione, da memorizzare nell'engine dell'Active Correlation Technology.

Fare riferimento a “Variabili” a pagina 24 per informazioni sulle variabili che possono essere utilizzate nelle espressioni. L'uso di certe variabili dipende dal contesto dell'espressione.

Attributi

<varInitializer> ha il seguente attributo:

Tabella 74. Attributi dell'elemento <varInitializer>

Nome	Descrizione	Tipo dati	Obbligatorio
expressionLanguage	Identifica il linguaggio di programmazione in cui viene scritta l'espressione. Poiché il linguaggio di programmazione Java è l'unico supportato, l'unico valore valido per questo attributo è java.	xsd:NMTOKEN	Yes

Contenuto in

<varInitializer> è contenuto nel seguente elemento:

- <variable>

Contiene

<varInitializer> non contiene elementi.

Concetti correlati

“Espressioni” a pagina 20

Un'espressione è un codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni possono anche accedere al codice esterno all'engine dell'Active Correlation Technology. Nel linguaggio della regola, le espressioni sono valide solo in contesti specifici o negli elementi di linguaggio della regola.

Elemento whenLoaded

L'elemento <whenLoaded> specifica che una regola viene attivata quando viene caricata dall'engine Active Correlation Technology.

Attributi

<whenLoaded> non ha attributi.

Contenuto in

<whenLoaded> è contenuto nel seguente elemento:

- <start>

Contiene

<whenLoaded> non contiene elementi.

Capitolo 6. Glossario

Questo glossario contiene i termini e le definizioni di concetti importanti dell'Active Correlation Technology.

ACT Vedi Active Correlation Technology.

azione Un'espressione eseguita come parte di una risposta di regola o quando una regola viene caricata, scaricata, attivata o disattivata.

Active Correlation Technology

Una tecnologia IBM che fornisce la correlazione eventi tramite le regole.

Compilatore dell'Active Correlation Technology

Il componente dell'Active Correlation Technology che analizza una serie di regole e il codice in essa contenuto per generare le strutture di dati interne richieste dall'engine Active Correlation Technology.

Engine dell'Active Correlation Technology

Il componente dell'Active Correlation Technology che elabora gli eventi in base all'output del compilatore dell'Active Correlation Technology.

Programma di creazione regole dell'Active Correlation Technology

Una GUI per scrivere le regole di correlazione nel linguaggio regole dell'Active Correlation Technology.

Linguaggio delle regole dell'Active Correlation Technology

Un linguaggio basato su XML per la scrittura dei regole per correlare gli eventi. Queste regole possono essere distribuite sugli ambienti di esecuzione dell'Active Correlation Technology.

Ambiente di esecuzione dell'Active Correlation Technology

Un'applicazione in cui è integrato l'engine dell'Active Correlation Technology con o senza il compilatore.

schema di raccolta

Uno schema di regola che definisce una regola per raccogliere un gruppo di eventi selezionati in un intervallo di tempo. Una regola che viene definita dallo schema di raccolta è una regola stateful.

schema di calcolo

Uno schema di regola che definisce una regola per applicare un calcolo (tramite un'espressione) agli eventi raccolti quando ogni evento viene ricevuto in un intervallo di tempo. Una regola che viene definita dallo schema di calcolo è una regola stateful.

dominio

Una categoria a cui si applica un gruppo di regole in base alla loro funzione. Ad esempio, un dominio può rappresentare un'area geografica specifica, la disciplina di gestione IT (come il rilevamento della sicurezza o la correlazione di eventi di rete) o l'organizzazione aziendale (come un'azienda o una sua divisione).

schema di duplicato

Uno schema di regola che definisce una regola che conta gli eventi secondi e successivi accettati all'interno di un intervallo di tempo specificato ma ignora l'elaborazione della serie di regole per questi eventi. Una regola che viene definita dallo schema di duplicato è una regola stateful.

fornitore di evento

Un software che genera gli eventi che vengono elaborati dall'Active Correlation Technology.

selettore eventi

I criteri per la selezione di eventi. Questi criteri determinano gli eventi che sono accettati per l'elaborazione della regola. Il selettore eventi include il tipo di evento e il predicato di filtro.

espressione

Il codice che contiene la logica personalizzata che può essere aggiunta ad una regola. Le espressioni vengono utilizzate per diversi scopi, quali l'inizializzazione delle variabili, la definizione dei criteri di selezione eventi o la specifica delle azioni di risposta di regola e di durata.

linguaggio dell'espressione

Il linguaggio di programmazione in cui viene scritta l'espressione.

evento esterno

Un evento che l'engine dell'Active Correlation Technology ricevuto da un'origine che è esterno ad esso.

oggetto esterno

Un oggetto che un'applicazione crea per comunicare con le espressioni.

predicato di filtro

Un'espressione che definisce la condizione in cui un evento è accettato per l'elaborazione della regola. Il predicato di filtro è una parte del selettore eventi. Un predicato di filtro restituisce un valore booleano.

schema di filtro

Uno schema di regola che definisce una regola per eseguire una certa azione quando accetta un evento. Una regola che è definita dallo schema di filtro agisce solo su un evento singolo ed è perciò una regola stateless.

chiave di raggruppamento

Un metodo per indirizzare una regola in modo da creare un'istanza della regola separata (o una sua copia) per ciascun gruppo di eventi che condivide caratteristiche comuni.

importa

L'importazione è un modo specifico del linguaggio di programmazione per rendere il codice esterno accessibile alle espressioni.

evento interno

Un evento che viene creato da una regola che è in esecuzione nell'engine dell'Active Correlation Technology. Questo evento può essere inoltrato ad altre regole.

azione di durata

Un'espressione eseguita quando una regola viene caricata, scaricata, attivata o disattivata.

nodo Un oggetto nella gerarchia di regole che può essere aggiunto, rimosso o sostituito, singolarmente e in modo autonomo, all'interno della serie di regole. I seguenti oggetti sono, ad esempio, nodi:

- Regole
- Blocchi di regole
- Variabili di blocco di regole
- Variabili di serie di regole

Poiché un oggetto non può essere elaborato singolarmente e in modo indipendente al di sotto del livello di regola, la variabile di regola non è un nodo.

predicato

Vedi predicato di filtro.

risposta

Vedi risposta di regola.

regola L'unità di correlazione che deve essere utilizzata per riconoscere le relazioni tra gli eventi e per eseguire le risposte di regola appropriate. Una regola è un'implementazione di uno dei seguenti sette degli schemi di regola ed è organizzata, in base alla propria funzione, in un blocco di regole che è parte di una serie di regole. Una regola accetta un evento per l'elaborazione se l'evento risponde ai criteri di selezione eventi.

blocco di regole

L'unità organizzativa per raggruppare le regole, in base alla funzione nei domini, all'interno della serie di regole. Un blocco di regole può contenere non solo regole ma anche altri blocchi di regole.

istanza della regola

Nel contesto della chiave di raggruppamento, una copia di una regola.

schema di regola

La rappresentazione di una condizione di correlazione eventi (come una condizione di soglia o un rilevamento di evento duplicato). Il linguaggio delle regole dell'Active Correlation Technology include i seguenti schemi di regola: raccolta, calcolo, duplicato, filtro, sequenza, soglia e timer. Lo schema di una regola ha valore quando si verifica la condizione definita dalla regola. Quando lo schema corrisponde, la regola conclude l'elaborazione eseguendo le azioni di risposta appropriate. Mentre è attiva una regola, lo schema di regola può essere applicato più volte.

risposta di regola

Un'espressione eseguita quando l'engine dell'Active Correlation Technology riconosce che è stata soddisfatta una condizione di regola. Una risposta di regola consiste di una o più azioni.

azione di risposta di regola

Vedi azione.

serie di regole

L'unità di esecuzione di regola per il linguaggio di regole dell'Active Correlation Technology. La serie di regole contiene le regole, organizzate in blocchi di regole, da eseguire tramite l'engine dell'Active Correlation Technology. L'engine agisce solo su una serie di regole per volta.

schema di sequenza

Uno schema di regola che definisce una regola per rilevare la presenza o assenza di eventi in un intervallo di tempo. La sequenza può essere ordinata o casuale. Una regola che viene definita dallo schema di calcolo è una regola stateful.

frammento

Un estratto di codice sorgente.

regola stateful

Una regola che contiene le informazioni di stato, che sono informazioni sulle caratteristiche di un'istanza della regola, allo scopo di agire su una

raccolta di eventi in un periodo di tempo. Le regole che sono definite da uno dei seguenti schemi di regola: raccolta, calcolo, duplicato, filtro, sequenza, soglia o timer.

regola stateless

Una regola che non contiene informazioni di stato e quindi può agire solo su un evento per volta. Una regola che è definita dallo schema di filtro è una regola stateless.

schema di soglia

Uno schema di regola che definisce una regola per raccogliere un gruppo di eventi selezionati in un intervallo di tempo e determina, dopo che è stato ricevuto ogni evento, se viene raggiunta una condizione di soglia. Una regola che viene definita dallo schema di soglia è una regola stateful.

schema timer

Uno schema di regola che definisce una regola per avviare azioni ad intervalli regolari. Una regola che viene definita dallo schema timer è una regola stateful. Sebbene una regola timer non elabora gli eventi, essa può essere attivata o disattivata da un evento.

Appendice. Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti. IBM può non offrire i prodotti, i servizi o le funzioni presentati in questo documento in altri paesi. Consultare il rappresentante locale IBM per informazioni sui prodotti ed i servizi attualmente disponibili nella propria zona. Qualsiasi riferimento ad un prodotto, programma o servizio IBM non implica o intende dichiarare che solo quel prodotto, programma o servizio IBM può essere utilizzato. E' possibile utilizzare, in alternativa, qualsiasi prodotto, programma o servizio funzionalmente equivalente che non violi alcun diritto di proprietà intellettuale IBM. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM dispone di applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura del presente documento non concede alcuna licenza a tali brevetti. Per informazioni relative a licenze rivolgersi per iscritto a:

IBM Director of Licensing
IBM Corporation
Shoenaicher D7030
Boeblingen Deutschland

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali.

IBM (INTERNATIONAL BUSINESS MACHINES CORPORATION) FORNISCE LA PRESENTE PUBBLICAZIONE NELLO STATO IN CUI SI TROVA, SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, MA NON LIMITATE A, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE.

Alcuni stati non consentono la rinuncia ad alcune garanzie espresse o implicite in determinate transazioni, pertanto, la presente dichiarazione può non essere applicabile.

Queste informazioni potrebbero includere inesattezze tecniche o errori tipografici. Le modifiche alle presenti informazioni vengono apportate periodicamente; tali modifiche saranno incorporate nelle nuove edizioni della pubblicazione. IBM può effettuare miglioramenti e/o modifiche ai prodotti e/o ai programmi descritti nella presente pubblicazione in qualsiasi momento senza preavviso.

Qualsiasi riferimento, contenuto in queste informazioni, a siti Web non IBM è fornito solo per comodità di consultazione e non conferisce, in alcun modo,

formale approvazione di tali siti Web. I materiali presenti in tali siti Web non fanno parte dei materiali previsti per questo prodotto IBM e pertanto l'utilizzo di tali siti Web è a proprio rischio.

IBM può utilizzare o distribuire qualsiasi informazione fornita in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

Coloro che detengono licenza d'uso di questo programma e che desiderano avere informazioni allo scopo di permettere: (i) lo scambio di informazioni tra i programmi creati indipendentemente e gli altri programmi (incluso il presente) e (ii) il reciproco utilizzo di informazioni che sono state scambiate, devono rivolgersi a:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Tali informazioni possono essere disponibili, in base ad appropriate clausole e condizioni, includendo in alcuni casi il pagamento di una tassa.

Il programma concesso in licenza descritto nel presente documento e tutto il materiale concesso in licenza disponibile sono forniti da IBM in base alle clausole dell'Accordo per Clienti IBM (IBM Customer Agreement), dell'IBM IPLA (IBM International Program License Agreement) o qualsiasi altro accordo equivalente tra le parti.

Marchi

DB2, IBM, il logo IBM, Tivoli, il logo Tivoli logo, Tivoli Enterprise Console e WebSphere sono marchi di International Business Machines Corporation negli Stati Uniti, in altri paesi o entrambi.

Java e tutti i marchi e i logo Java sono marchi di Sun Microsystems, Inc. negli Stati Uniti, in altri paesi o entrambi.

Nomi di altre società, prodotti o servizi possono essere marchi di altre società.



Stampato in Italia